

Stanovení mědi metodou kalibrační přímky

Ze zásobního roztoku modré skalice o koncentraci Cu^{2+} $1 \text{ g}\cdot\text{l}^{-1}$ postupným ředěním připravíme výchozí roztok o koncentraci Cu^{2+} $1 \text{ mg}\cdot\text{l}^{-1}$. Z tohoto roztoku připravíme kalibrační roztoky o koncentracích Cu^{2+} v $\text{mg}\cdot\text{l}^{-1}$: 0; 0,02; 0,04; 0,06; 0,08; 0,10; 0,12 do 25ml odměrných baněk. Baňky doplníme po rysku 1% HNO_3 . Vzorek bílého vína 5krát naředíme, tedy do 25 ml baňky nepipetujeme 5 ml bílého vína a baňku doplníme po rysku 1% HNO_3 . Takto upravený vzorek následně rozdělíme do tří baněk. Jako blank použijeme roztok o koncentraci $0 \text{ mg}\cdot\text{l}^{-1}$. Pro všechny připravené roztoky 7krát proměříme absorbanci.

Výsledky přepíšeme do samostatných sloupců v Excelu a uložíme jako txt soubor. **Při přepisování dat, se kterými budeme dále počítat, dbáme na to, abychom místo desetinné čárky vždy používali tečky. Pouze tento formát je R Studio schopné načíst a dále s ním pracovat.** Při ukládání souboru do formátu txt je vždy ukládán jen aktivní list, tedy ten, s kterým právě pracujeme. Proto je nutné, aby na každém listě byla data pro jednotlivé operace samostatně. Na první list přepíšeme data z měření pro kalibrační přímku a uložíme pod názvem **kal_primka** do složky **kalibrace**, která je umístěna na ploše počítače. Jestli je ve složce soubor se stejným názvem, nahradíme ho novým. Následující tabulka zobrazuje vzorová data:

Tab. č. 4: Vzorová data získaná z měření metodou kalibrační přímky

0 mg/l	0,02 mg/l	0,04 mg/l	0,06 mg/l	0,08 mg/l	0,10 mg/l	0,12 mg/l
0.003258	0.006297	0.008151	0.01138	0.01449	0.01878	0.02079
0.002864	0.006751	0.008377	0.01154	0.01509	0.01836	0.02126
0.002883	0.006925	0.008282	0.01164	0.01534	0.01846	0.02101
0.002908	0.006433	0.008075	0.01174	0.01392	0.01852	0.02051
0.003203	0.006288	0.008510	0.01148	0.01427	0.01853	0.02050
0.002847	0.006505	0.008116	0.01145	0.01483	0.01835	0.02079
0.002902	0.006526	0.008138	0.01110	0.01440	0.01892	0.02047

Pro vzorky postupujeme obdobně. Na nový list v Excelu přepíšeme hodnoty absorbance z měření roztoků vzorků a uložíme jako **vzorky** do složky **kalibrace**. Postupujeme stejně jako s předchozím souborem. V tabulce níže můžeme vidět příklad dat naměřených pro vzorky:

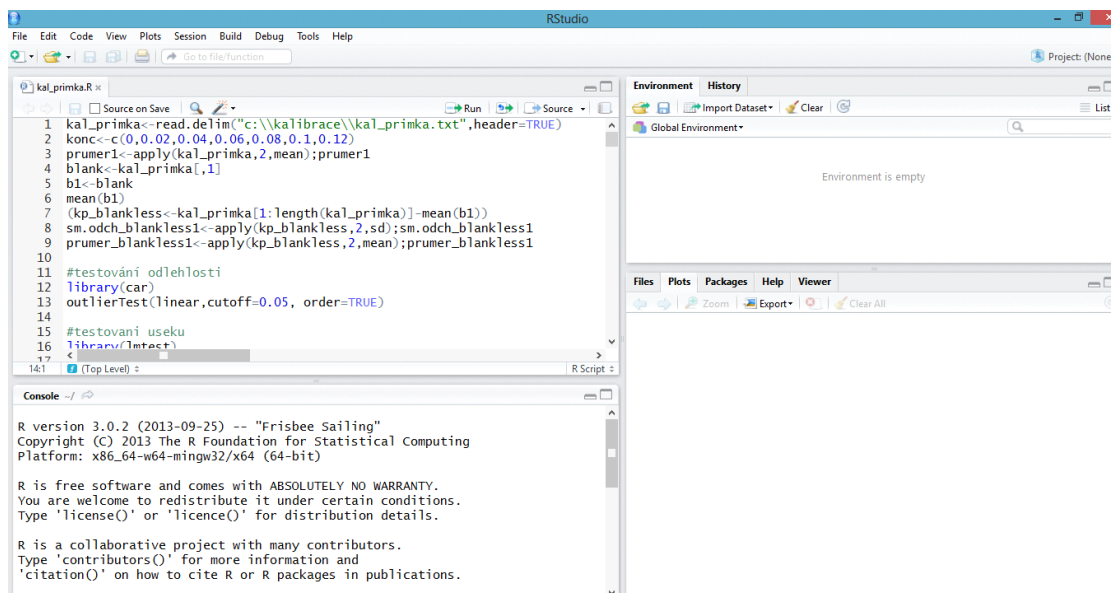
Tab. č. 5: Data získaná z měření vzorků

vz1	vz2	vz3
0.007447	0.008718	0.008233

0.007573	0.008035	0.008679
0.007191	0.008009	0.008470
0.007579	0.008235	0.008254
0.007871	0.008047	0.008430
0.007421	0.008390	0.008551
0.007442	0.007985	0.008468

Nyní můžeme začít se zpracováním v R Studiu:

Zapneme R Studio, pokud není otevřen soubor kal_primka.R, klikneme na „File“, „Open File“ a ze složky kalibrace vybereme soubor kal_primka.R. Po otevření souboru by mělo okno vypadat zhruba takto:



Obrázek č. 6: Otevřené R Studio

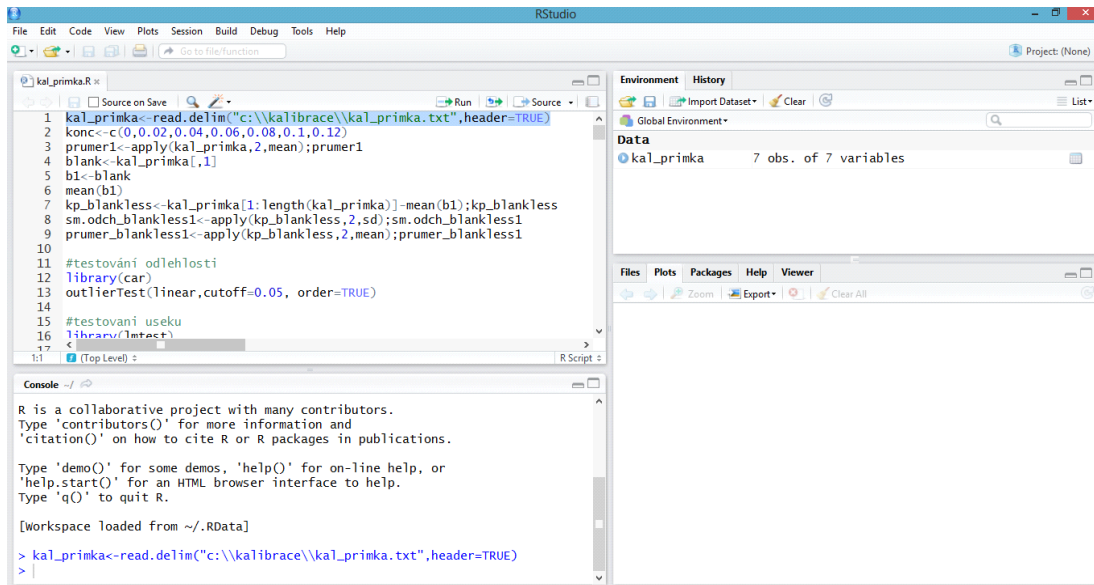
Vlevo nahoře vidíme příkazový řádek, pod ním konzolu, která odpovídá na jednotlivé příkazy, vpravo nahoře vidíme prostředí, ve kterém se ukládají jednotlivé hodnoty, data a vypočítané hodnoty, a nakonec vpravo dole vidíme okno se záložkami, které zobrazuje grafy a pomocí kterého můžeme kontrolovat jednotlivé knihovny.

Myší modře označíme první řádek skriptu

- *kal_primka<-read.delim("c:\\kalibrace\\kal_primka.txt",header=TRUE)*

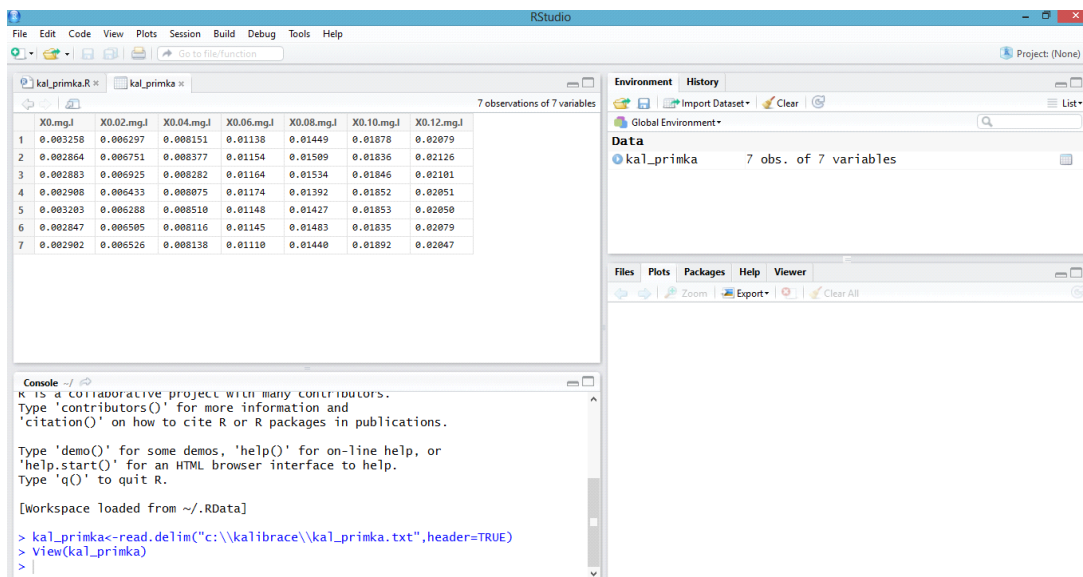
a klikneme na RUN nebo zmáčkneme Ctrl+R (enter nefunguje). Tím načteme uložená data. Na konci skriptu je napsáno „header=TRUE“, pokud jsme data v Excelu ukládali bez popisek, tedy bez hlavičky, TRUE přepíšeme na FALSE. V tomto případě je nutné psát velkými

písmeny. Dle umístění se červeně psaná část skriptu může lišit. Pokud skript funguje, zobrazí se modře na konzole a v prostředí se uloží načtená data.



Obrázek č. 7: Spuštění prvního řádku skriptu

Kliknutím na název v prostředí se data zobrazí:



Obrázek č. 8: Zobrazení načtených dat

Označíme další řádek, který udává vektor koncentrace. Pokud se koncentrace liší, jednoduše hodnoty přepíšeme.

- `konc<-c(0,0.02,0.04,0.06,0.08,0.1,0.12);konc`

Zkontrolujeme data zobrazená na konzole, pokračujeme dalším řádkem, pomocí kterého první sloupec z původních dat označíme jako blank.

- `blank<-kal_primka[,1];blank`

Z toho vypočítáme průměr

- `mean(blank)`

a tento průměr odečteme od celé matice, abychom dále pracovali hodnotami naměřené absorbance korigované na blank.

- `kp_blankless<-kal_primka[1:length(kal_primka)]-mean(blank);kp_blankless`

Zobrazenou matici zkontrolujeme na konzole, nebo kliknutím na název v prostředí. Nakonec vypočítáme z každého sloupce průměr, který použijeme pro zobrazení průměrných hodnot v grafu kalibrační přímky

- `prumer_blankless1<-apply(kp_blankless,2,mean);prumer_blankless1`

Nyní musíme data otestovat na odlehlost pomocí Grubbsova testu, ale testujeme data pro lineární závislost, takže napřed musíme tuto závislost vytvořit. V následujícím skriptu se vstupní data uspořádají do dvou sloupců a díky tomu budeme dále schopni udělat regresi pro všechna vstupní data. Označíme následující řádky a spustíme:

- `a<-length(kp_blankless[1,]);a`
- `b<-length(kp_blankless[,1]);b`
- `X<-rep(konc,b);X`
- `Y<-kp_blankless[1,]`
- `for(i in 2:b){`
`YY<-kp_blankless[i,]`
`Y<-as.numeric(c(Y,YY))};Y`

Označíme další řádek, kterým dáme R Studiu příkaz, že data jsou lineárního charakteru:

- `linear<-lm(Y~X);linear`

V okně konzoly se zobrazí hodnota úseku (intercept) a směrnice (konc). Abychom mohli provést testování odlehlosti, musíme „zavolat“ potřebnou knihovnu.

- `library(car)`

V další kroku otestujeme data:

- `outlierTest(linear,cutoff=0.05, order=TRUE)`

Pokud jsou data v pořádku, zobrazí se v okně konzoly:

No studentized residuals with Bonferroni $p < 0,05$

Ale zobrazí se nejodlehlejší hodnota. Pokud se zobrazí nějaká odlehlá hodnota, poznamenáme si ji a uvedeme do protokolu.

Dále provedeme testování úseku a směrnice. Musíme aktivovat potřebnou knihovnu, ze které budeme používat funkci

- `library(lmtest)`

Pravděpodobně se zobrazí varovné hlášení, toto hlášení můžeme ignorovat.

```
Loading required package: zoo
```

```
Attaching package: 'zoo'
```

```
The following objects are masked from 'package:base':
```

```
as.Date, as.Date.numeric
```

```
Warning messages:
```

```
1: package 'lmtest' was built under R version 3.0.3  
2: package 'zoo' was built under R version 3.0.3
```

Dále můžeme provést samotné testování úseku a směrnice, předpokládáme nulovou hypotézou $a = 0$ a $b = 1$:

- `coeftest(linear, vcov. = NULL, df = NULL)`

Zobrazí se data:

```
t test of coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-7.6592e-05	3.8018e-04	-0.2015	0.8483
konc	1.4961e-01	5.2721e-03	28.3778	1.018e-06 ***

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

„Intercept“ značí úsek, „konc“ směrnici, „Estimate“ znamená odhad, „Std. Error“ udává směrodatnou odchylku, „t value“ je vypočtená kritická hodnota Studentova rozdělení a „Pr(<|t|)“ udává pravděpodobnost v intervalu $<0;1>$, že nulová hypotéza pro úsek ($a = 0$) a pro směrnici ($b = 1$) je pravdivá.

Dle pravděpodobnosti, která vyjde pro hypotézu $a = 0$, se dále musíme rozhodnout, zda úsek zanedbáme. Pokud vyjde $\text{Pr}(>|t|) > 0.50$, pokračujeme bodem 1, pokud $\text{Pr}(>|t|) < 0.50$, bod 1 přeskočíme a pokračujeme bodem 2 (řádek skriptu kolem 135).

1.

1. $\text{Pr}(>|t|) > 0.50$, pak přijmeme nulovou hypotézu pro úsek ($a = 0$) a dále operujeme s přímkou, která prochází počátkem.

Do příkazového řádku tedy zadáme $y = 0 + b \cdot x$ pro vlastní data:

- `linear1<-lm(Y~0+X);linear1`

Zobrazí se **hodnota úseku**, tu si poznamenejme a při tvorbě grafu toto číslo zadáme do legendy. Nyní můžeme přejít k tvorbě grafu, označíme další řádek a klikneme na RUN

- `plot(X,Y,type="n",main="Kalibrační přímka", xlab= expression(Koncentrace ~ Cu ~ (mg ~ "*" ~ l^{-1})),ylab="Absorbance (a.u.)",col=3)`

Zatím se žádný graf nezobrazí, protože R Studio nyní jen připravilo data. Ovšem v pravém dolním okně se v záložce „Plots“ objeví název grafu a souřadný systém. Vzhledem k tomu, že rámeček, který ohraničuje legendu, je neprůhledný a překrýval by přímku, musíme nejprve zadat legendu, až poté přidat body („points“) a samotnou přímku („abline“):

- `legend(-0.003,0.018,legend=c("kalibrační přímka - y = 0,1487x","chybové úsečky - int. spolehlivosti"),col=c(4,6), box.lwd=0, box.col="white", bg="white", lty=c(1,1), cex=0.9, lwd=1.5, pch=c("-", "-"))`

Místo červeně napsané hodnoty směrnice zadáme vlastní. Pokud rámeček legendy překrývá osy, jeho polohu můžeme upravit změnou čísel v závorce `legend(-0.003,0.018`, tyto souřadnice udávají polohu levého horního rohu bílého rámečku legendy. Po úpravě hodnot musíme celý skript spustit znovu, a to nejen pro legendu, ale i pro graf (`plot...`).

Do grafu nezobrazíme všechny body, ale jen jejich průměrné hodnoty, aby byly chybové úsečky dobře viditelné:

- `points(konc,prumer_blankless1,col=1)`
- `abline(linear1,col=4)`

Nyní se zobrazí graf, do toho ale ještě přidáme chybové úsečky. Ty zobrazujeme pomocí intervalu spolehlivosti. Na jeho výpočet potřebujeme směrodatnou odchylku a průměr pro každý bod kalibrační přímky, ten už máme, takže dopočítáme pouze směrodatné odchylky:

- `sm.odch_blankless1<-apply(kp_blankless,2,sd);sm.odch_blankless1`

Dále je třeba vypočítat hodnoty, které budeme odečítat a přičítat k průměru, abychom vypočítali intervaly pro chybové úsečky, což je směrodatná odchylka vynásobená kritickou hodnotou Studentova rozdělení pro $v = n - 1$, to celé poděleno odmocninou z n , což je počet měření pro každý kalibrační bod.

- `L1<-((sm.odch_blankless1*qt(0.975,6))/sqrt(length(blank)));L1`

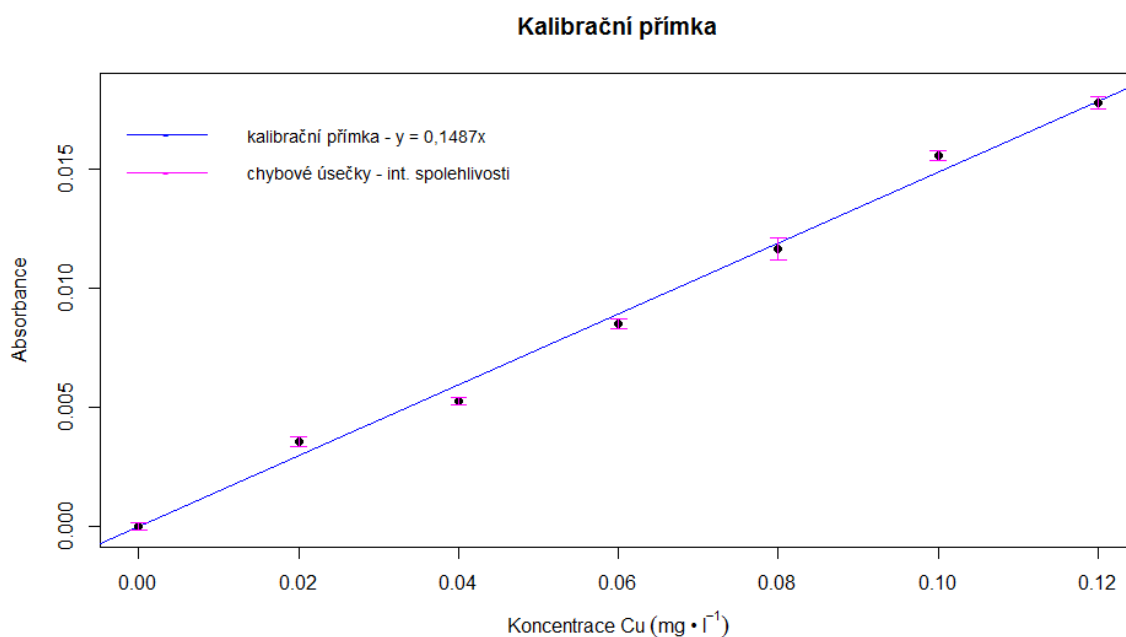
Pro zobrazení chybových úseček potřebujeme aktivovat knihovnu

- `library(chemCal)`

a nyní konečně můžeme zobrazit chybové úsečky pomocí skriptu:

- `errbar(konc, prumer_blankless1, yplus=(prumer_blankless1+L1), yminus=(prumer_blankless1-L1), cap=0.015, add=T, errbar.col=6)`

V grafu se zobrazí chybové úsečky. Pro uložení grafu klikneme na „Zoom“ a na hlavním panelu počítače se otevře okno s názvem „Plot Zoom“, ve kterém se zobrazí obrázek grafu. Klikneme na něj pravým tlačítkem myši a zvolíme „Save Image“, graf přiložíme do protokolu.



Obrázek č. 9: Graf kalibrační přímky se zobrazenými chybovými úsečkami

V dalším grafu zobrazíme pás spolehlivosti a predikční pás, musíme spustit knihovnu, ze které použijeme funkci:

- `library(chemCal)`

a také zdrojový kód, ve kterém byla upravena legenda a popisky os pro potřeby této úlohy.

Veškeré potřebné parametry už jsou předdefinovány

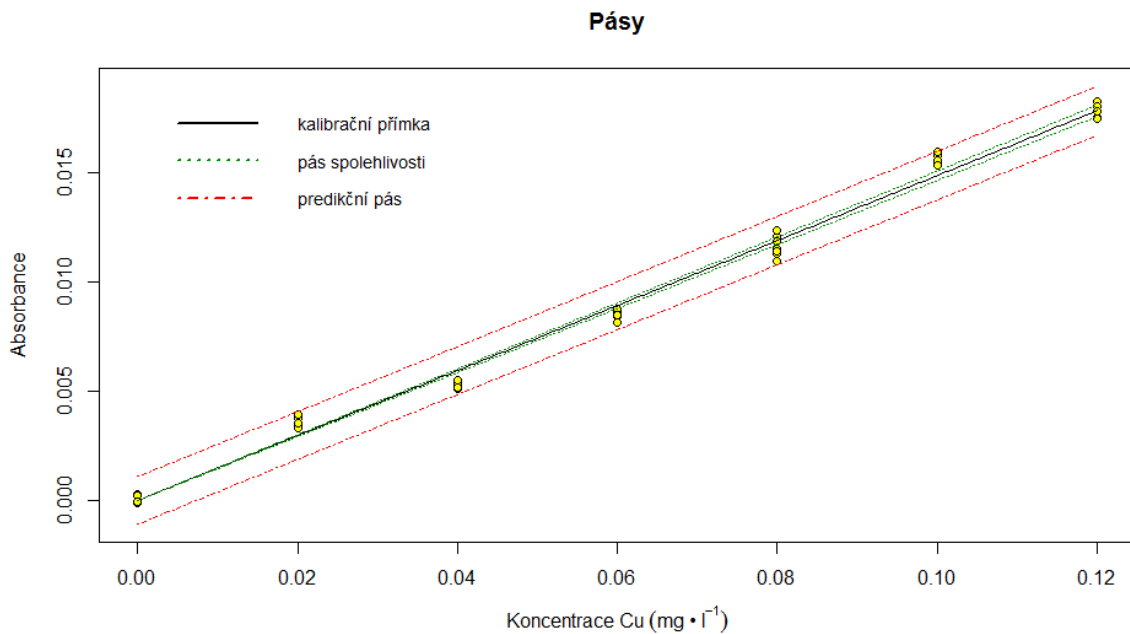
- `source("C:\\kalibrace\\calplot_primka.r")`

Nyní můžeme vytvořit graf se zmíněnými pásy

- `linear2<-calplot_primka(linear1, xlim = c("auto", "auto"), ylim = c("auto", "auto"), xlab=expression(Koncentrace ~ Cu ~ (mg ~ "*" ~ l^{-1})) ylab="Absorbance (a.u.) ", alpha=0.05, varfunc = NULL)`

Zobrazí se graf. Pro jeho uložení opět klikneme na „Zoom“ a pravým tlačítkem myši na obrázek. Obrázek přiložíme k protokolu. V grafu si můžeme všimnout, že hyperbola, která

určuje pás spolehlivosti, se sbíhá v nule, protože jsme definovali, že přímka prochází přesně počátkem.



Obrázek č. 10: Graf pásu spolehlivosti a predikčního pásu

Dále vypočítáme intervaly spolehlivosti pro parametry, v tomto případě pouze pro směrnici. K tomu budeme potřebovat směrodatnou odchylku pro směrnici. Tu zjistíme pomocí funkce „coefstest“ pro definovanou kalibrační přímku

- `coefstest(linear1, vcov. = NULL, df = NULL)`

Zobrazí se data

t test of coefficients:

```

      Estimate Std. Error t value Pr(>|t|)
konc 0.1487275  0.0026805  55.486 2.301e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Abychom mohli se zjištěnými hodnotami dále pracovat, musíme je zobrazit jako tabulku. To uděláme pomocí funkce „as.matrix“:

- `pint1<-as.matrix(coefstest(linear1, vcov. = NULL, df = NULL))`

Z tabulky vyčleníme jednotlivé hodnoty. `pint1_b1` reprezentuje úsek a `pint1_b2` reprezentuje směrodatnou odchylku úseku:

- `pint1_b1<-pint1[1,1];pint1_b1`
- `pint1_b2<-pint1[1,2];pint1_b2`

Směrodatnou odchylku zaokrouhlíme na 2 platné číslice a podle počtu míst za desetinnou tečkou poté zaokrouhlíme výsledný interval spolehlivosti:

- `pb2<-signif(pint1_b2, digits=2);pb2`

dostaneme např: 0.0027, tedy 4 desetinná místa

Nyní můžeme zjistit samotný interval:

- `L1<-c(pint1_b1-pb2*qt(0.975,5),pint1_b1+pb2*qt(0.975,5));L1`

Interval zaokrouhlíme na platný počet míst:

- `L1b<-round(L1, digits=4);L1b`

Zobrazený interval uvedeme do protokolu: $L_{(b)1,2} = \langle 0.1418; 0.1557 \rangle$.

Nyní vypočítáme mez detekce a mez stanovitelnosti

- `lod(linear1)`
- `ld<-as.numeric(lod(linear));ld`

```
[1] 0.012252509 0.001822285
```

- `lq<-(ld*10/3);lq`

```
[1] 0.040841697 0.006074282
```

Mez detekce i stanovitelnosti se odečítá na ose x, takže nás v obou případech zajímá první hodnota, což je koncentrace, zjištěné hodnoty si zapíšeme a poznamenejeme v protokolu.

(ld = mez detekce, lq = mez stanovitelnosti)

Nyní zobrazíme elipsu spolehlivosti a predikce. Pro to budeme potřebovat další knihovnu

- `library(ellipse)`

Poté můžeme spustit skript pro elipsy. Celý skript označíme myší a spustíme. Tím vypočítáme všechna potřebná data pro vynesení elips do grafu:

- `level <- 0.95`
- `shape <- var(cbind(X, Y))`
- `center <- c(mean(konc), mean(prumer_blankless1))`
- `radiusC <- sqrt(2 *(length(konc)-1)* qf(level, 2, (length(konc) - 2))/(length(konc)*(length(konc)-2)))`
- `radiusP <- radiusC*sqrt(length(konc)+1)`

- `ellcalc <- function(center, shape, radius, segments=1000){`
 `segments=segments`
 `angles <-<- seq(from=0,to=(2*3.1415926),by=1/segments)`

```
unit.circle <- cbind(cos(angles), sin(angles))
ELLIPSE <- t(center + radius * t(unit.circle %*% chol(shape)))
return(ELLIPSE)}
```

- `ellC <- ellcalc(center, shape, radiusC)`
- `ellP <- ellcalc(center, shape, radiusP)`

Pokud je vše v pořádku, můžeme přistoupit k tvorbě grafu. Zadáme příkaz ke zpracování dat:

- `plot(X,Y, type="n", main="elipsy", xlab=expression(Koncentrace ~ Cu ~ (mg ~ "*" ~ l^{-1})), ylab="Absorbance (a.u.)", xlim=c(-0.12,0.25), ylim=c(-0.02,0.04))`

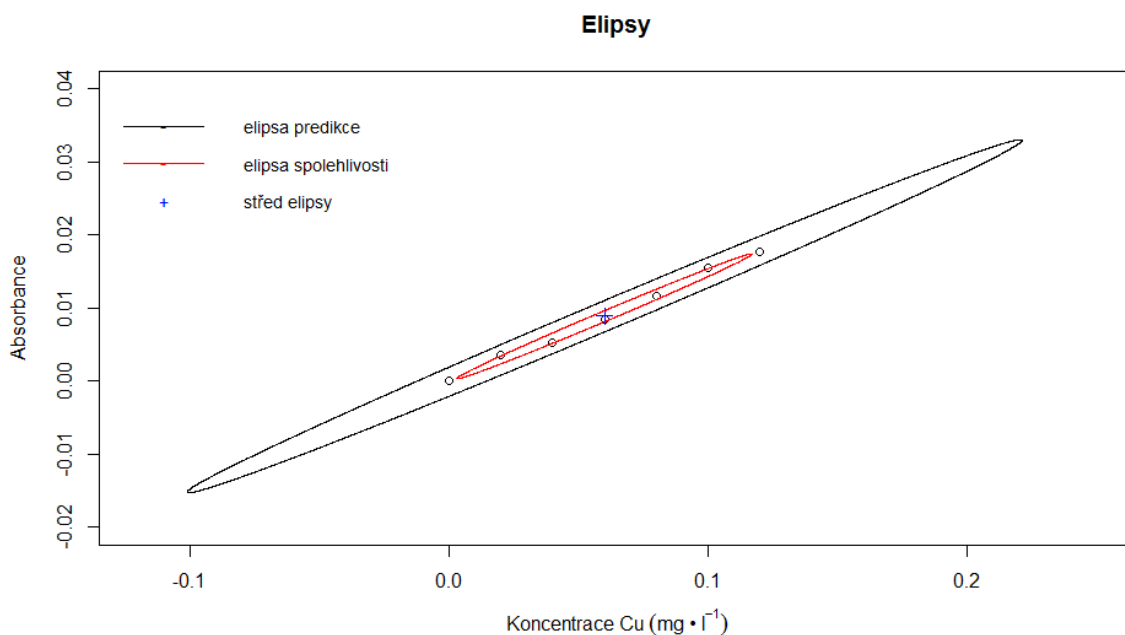
Pokud není elipsa vidět celá, změníme úsek, který chceme na osách zobrazit. Toho docílíme změnou čísel v závorce u `xlim=c(-0.12,0.25)` a `ylim=c(-0.02,0.04)`. Hodnoty v závorce udávají počátek a konec osy. Dále zobrazíme legendu:

- `legend(-0.13,0.04,c("elipsa predikce", "elipsa spolehlivosti", "průměr"), col=c(1,2,4), lty=c(1,1,0), box.lwd = 0, box.col = "white", bg = "white", cex=0.9, lwd=1.5, pch=c("-", "-", "+"))`

Pozici legendy opět můžeme upravit změnou čísel v závorce `legend(-0.13,0.04)`. V grafu zobrazíme průměrné hodnoty bodů kalibrační přímky, střed elipsy a poté samotné elipsy:

- `points(konc, prumer_blankless, col=1)`
- `points(center[1], center[2], pch = 3, col=4, cex=1.25)`
- `points(ellC[,1], ellC[,2], col=2, type="l")`
- `points(ellP[,1], ellP[,2], col=1, type="l")`

Graf zobrazíme stejně jako předchozí, uložíme a přiložíme do protokolu.



Graf č. 11: Elipsa spolehlivosti a predikce

Nyní přikročíme k výpočtu koncentrace mědi ve vzorku. Načteme soubor, do kterého jsme uložili naměřené signály pro vzorky:

- `vzorky<-read.delim("c:\\kalibrace\\vzorky.txt",header=TRUE)`

Z každého sloupce vypočítáme průměr a v dalším kroku odečteme blank

- `prumer3<-apply(vzorky,2,mean);prumer3`
- `vzorky_blankless<-prumer3[1:length(prumer3)]-mean(blank);vzorky_blankless`

Zkontrolujeme zobrazené hodnoty. Dále vypočítáme koncentraci mědi, a to tak, že průměrné hodnoty signálu podělíme směrnici přímky

- `konc_Cu<-((vzorky_blankless/pint1_b1)*5);konc_Cu`

Jednotlivé vypočtené koncentrace zobrazíme každou samostatně opět pomocí funkce `as.matrix`

- `vzorky_Cu<-as.matrix(konc_Cu);vzorky_Cu`
- `vz1<-vzorky_Cu[1,1];vz1`
- `vz2<-vzorky_Cu[2,1];vz2`
- `vz3<-vzorky_Cu[3,1];vz3`

Poté vypočítáme intervaly spolehlivosti pro jednotlivé vzorky. K samotnému výpočtu budeme potřebovat průměrnou hodnotu x a průměrnou hodnotu y z kalibrační přímky:

- `xpr<-mean(konc);xpr`
- `ypr<-mean(prumer_blankless1);ypr`

Dále hodnoty y pro každou koncentraci vypočtené z regresní přímky:

- `Y<-(0+konc*pint1_b1);Y`

počet všech naměřených bodů použitých pro kalibraci

- `n<-(length(konc)*length(blank));n`

a také kolikrát byl změřen signál pro každý vzorek:

- `m<-length(vzorky[,1]);m`

Abychom vypočítali, $s'_{y,x}$ potřebujeme znát sumu druhých mocnin z rozdílů všech naměřených a vypočítaných hodnot pro jednotlivé koncentrace kalibračních roztoků z kalibrační přímky. Musíme postupovat tak, že z původní matice naměřených hodnot po odečtení blanku vyčleníme jednotlivé řádky, od každého odečteme vektor vypočítaných hodnot, umocníme, nakonec sečteme. Označíme celý skript a spustíme

- $kp1 <- kp_blankless[1,]; kp1$
- $kp2 <- kp_blankless[2,]; kp2$
- $kp3 <- kp_blankless[3,]; kp3$
- $kp4 <- kp_blankless[4,]; kp4$
- $kp5 <- kp_blankless[5,]; kp5$
- $kp6 <- kp_blankless[6,]; kp6$
- $kp7 <- kp_blankless[7,]; kp7$
- $kpY1 <- -(sum((kp1-Y)^2)); kpY1$
- $kpY2 <- -(sum((kp2-Y)^2)); kpY2$
- $kpY3 <- -(sum((kp3-Y)^2)); kpY3$
- $kpY4 <- -(sum((kp4-Y)^2)); kpY4$
- $kpY5 <- -(sum((kp5-Y)^2)); kpY5$
- $kpY6 <- -(sum((kp6-Y)^2)); kpY6$
- $kpY7 <- -(sum((kp7-Y)^2)); kpY7$
- $kpY <- (kpY1+kpY2+kpY3+kpY4+kpY5+kpY6+kpY7); kpY$

Nyní můžeme dopočítat $s'_{y,x}$

- $syxp <- sqrt((kpY)/(n-1)); syxp$

a poté směrodatné odchylky pro jednotlivé vzorky

- $sx1 <- ((syxp/pint1_b1)*sqrt((1/m)+(pb2/(pint1_b1*syxp))^2*vz1^2)); sx1$
- $sx2 <- ((syxp/pint1_b1)*sqrt((1/m)+(pb2/(pint1_b1*syxp))^2*vz2^2)); sx2$
- $sx3 <- ((syxp/pint1_b1)*sqrt((1/m)+(pb2/(pint1_b1*syxp))^2*vz3^2)); sx3$

Ty zaokrouhlíme na dvě platné číslice a podle počtu míst za desetinnou tečkou pak zaokrouhlíme jednotlivé intervaly:

- $sxp1 <- signif(sp1, digits=2); sxp1$
- $sxp2 <- signif(sp2, digits=2); sxp2$
- $sxp3 <- signif(sp3, digits=2); sxp3$

Nakonec vypočítáme intervaly spolehlivosti pro jednotlivé vzorky

- $Lvz1 <- c(vz1-sxp1*qt(0.975,48), vz1+sxp1*qt(0.975,48)); Lvz1$
- $Lvz2 <- c(vz2-sxp2*qt(0.975,48), vz2+sxp2*qt(0.975,48)); Lvz2$
- $Lvz3 <- c(vz3-sxp3*qt(0.975,48), vz3+sxp3*qt(0.975,48)); Lvz3$

a zaokrouhlíme na odpovídající počet platných desetinných míst:

- $Lvz1 <- round(Lvz1, digits=4); Lvz1$
- $Lvz2 <- round(Lvz2, digits=4); Lvz2$
- $Lvz3 <- round(Lvz3, digits=4); Lvz3$

Intervaly spolehlivosti stanoveného množství Cu^{2+} si poznamenejme a uvedeme do protokolu jako

$$\text{Lvz1} = \langle 0.1365; 0.1675 \rangle \text{ mg} \cdot \text{l}^{-1}$$

$$\text{Lvz2} = \langle 0.1578; 0.1933 \rangle \text{ mg} \cdot \text{l}^{-1}$$

$$\text{Lvz3} = \langle 0.1650; 0.2021 \rangle \text{ mg} \cdot \text{l}^{-1}$$

2.

1. $\text{Pr}(|t| < 0.50)$, pak zamítneme nulovou hypotézu pro úsek ($a = 0$) a dále operujeme s přímkou, která neprochází počátkem.

Do příkazového řádku tedy zadáme $y = a + b \cdot x$ pro vlastní data:

- `linear3<-lm(Y~X);linear3`

Zobrazí se **hodnota úseku a směrnice**, ty si poznamenejme a při tvorbě grafu tato čísla zadáme do legendy. Nyní můžeme přejít k tvorbě grafu, označíme další řádek a klikneme na RUN

- `plot(konc,prumer_blankless1,type="n",main="Kalibrační přímka", xlab= expression (Koncentrace ~ Cu ~ (mg ~ "*" ~ l^{-1})),ylab="Absorbance (a.u.)", col=3)`

Zatím se žádný graf nezobrazí, R Studio jen připravilo data. Ovšem v pravém dolním okně se v záložce „Plots“ objeví název grafu a osy. Vzhledem k tomu, že rámeček legendy je neprůhledný a překrýval by přímkou, musíme nejprve zadat legendu, až poté přidat body (points) a samotnou přímkou (abline):

- `legend(-0.003,0.018,legend=c(expression(kal. ~ přímka ~ y ~ "-7,659" ~ "*" ~ 10^{-5} ~ "+" ~ "1,496" ~ "*" ~ 10^{-1}~x), "chybové úsečky - int. spolehlivosti"), col=c(4,6),box.lwd = 0,box.col = "white",bg = "white", lty=c(1,1), cex=0.9, lwd=1.5,pch=c("-", "-"))`

Místo červeně napsaných hodnot zadáme vlastní. Pokud rámeček legendy překrývá osy, můžeme její pozici upravit změnou čísel v závorce `legend(-0.003,0.018`, tyto souřadnice udávají polohu levého horního rohu bílého rámečku legendy. Pokud je rámeček legendy příliš dlouhý, bohužel bude překrývat orámování grafu. Po úpravě hodnot musíme celý skript spustit znovu, a to nejen pro legendu, ale i pro graf (`plot...`). Přidáme body a samotnou přímkou:

- `points(konc,prumer_blankless1,col=1)`
- `abline(linear3,col=4)`

Nyní se zobrazí graf, do toho ale ještě přidáme chybové úsečky. Ty zobrazujeme pomocí intervalu spolehlivosti. Na jeho výpočet potřebujeme směrodatnou odchylku a průměr pro každý bod kalibrační přímky, ten už máme, takže dopočítáme pouze směrodatné odchylky:

- `sm.odch_blankless1<-apply(kp_blankless,2,sd);sm.odch_blankless1`

Dále je třeba vypočítat hodnoty, které budeme odečítat a přičítat k průměru, abychom vypočítali intervaly pro chybové úsečky, což je směrodatná odchylka vynásobena pro $v = n - 1$, to celé poděleno odmocninou z n , což je počet měření pro každý kalibrační bod.

- `L1<-((sm.odch_blankless1*qt(0.975,6))/sqrt(length(blank)));L1`

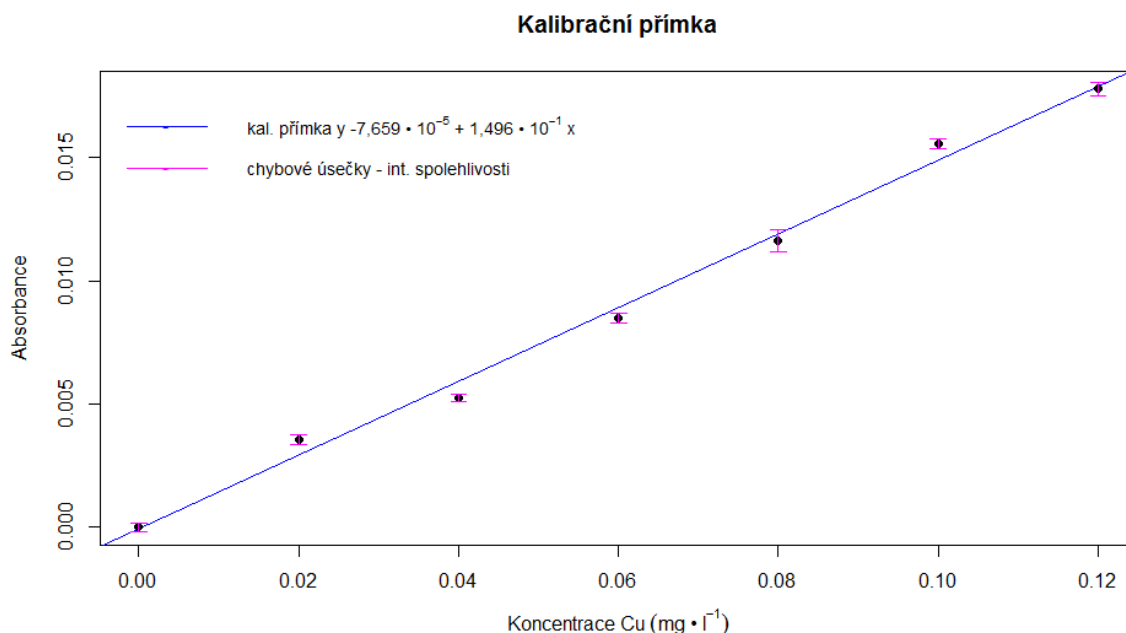
Pro zobrazení chybových úseček potřebujeme přivolat knihovnu

- `library(chemCal)`

a nyní můžeme zobrazit chybové úsečky pomocí skriptu:

- `errbar(konc, prumer_blankless1, yplus=(prumer_blankless1+L1),
yminus=(prumer_blankless1-L1), cap=0.015, add=T, errbar.col=6)`

V grafu se zobrazí chybové úsečky. Pro uložení grafu klikneme na „Zoom“ a na hlavním panelu počítače se otevře okno s názvem „Plot Zoom“, ve kterém se zobrazí obrázek grafu. Klikneme na něj pravým tlačítkem myši a zvolíme „Save Image“, přidáme do protokolu.



Obrázek č. 9: Graf kalibrační přímky se zobrazenými chybovými úsečkami

V dalším grafu zobrazíme pásy spolehlivosti a predikce, musíme otevřít knihovnu, ze které použijeme funkci:

- `library(chemCal)`

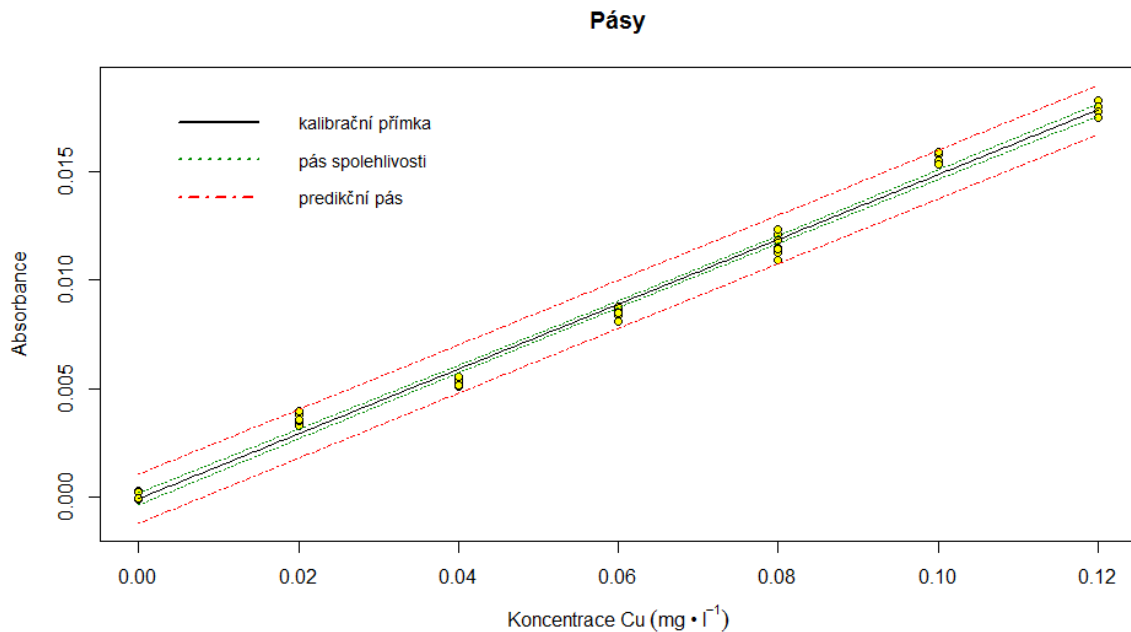
a také zdrojový kód, který byl předem upraven pro potřeby této úlohy. Veškeré potřebné parametry už jsou předdefinovány

- `source("C:\\kalibrace\\calplot_primka.r")`

Nyní můžeme vytvořit graf se zmíněnými pásy

- `linear4<-calplot_primka(linear3, xlim = c("auto", "auto"), ylim = c("auto", "auto"), xlab=expression(Koncentrace ~ Cu ~ (mg ~ "*" ~ l^{-1})), ylab= "Absorbance (a.u.)", alpha=0.05, varfunc = NULL)`

Zobrazí se graf. Pro jeho uložení opět klikneme na „Zoom“ a pravým tlačítkem myši na obrázek. Obrázek přiložíme k protokolu.



Obrázek č. 10: Graf pásu spolehlivosti a predikčního pásu

Dále vypočítáme intervaly spolehlivosti pro parametry. K tomu budeme potřebovat směrodatnou odchylku pro úsek a směrnici. Ty zjistíme pomocí funkce „`coefstest`“ pro definovanou kalibrační přímku

- `coefstest(linear3, vcov. = NULL, df = NULL)`

Zobrazí se data

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-7.6592e-05	3.8018e-04	-0.2015	0.8483
konc	1.4961e-01	5.2721e-03	28.3778	1.018e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Abychom mohli se zjištěnými hodnotami dále pracovat, musíme je zobrazit jako tabulku. To uděláme pomocí funkce „as.matrix“:

- `pint1<-as.matrix(coeftest(linear1, vcov. = NULL, df = NULL))`

Jednotlivé hodnoty zobrazíme samostatně. `Pint1_a1` reprezentuje úsek, `Pint1_a2` značí směrodatnou odchylku úseku, `Pint1_b1` reprezentuje úsek a `pint1_b2` udává směrodatnou odchylku úseku. Můžeme spustit všechny řádky současně:

- `pint1_a1<-pint1[1,1];pint1_a1`
- `pint1_a2<-pint1[1,2];pint1_a2`
- `pint1_b1<-pint1[2,1];pint1_b1`
- `pint1_b2<-pint1[2,2];pint1_b2`

Směrodatné odchylky zaokrouhlíme na dvě platné číslice a podle počtu desetinných míst u výsledné hodnoty zaokrouhlíme vypočítané intervaly:

- `pa2<-signif(pint1_a2, digits=2);pa2`
- `pb2<-signif(pint1_b2, digits=2);pb2`

Dostaneme např: `pa2 = 0.00014` a `pb2 = 0.0020`

Nyní můžeme zjistit samotné intervaly:

- `La<-c(pint1_a1-pint1_a2*qt(0.975,5),pint1_a1+pint1_a2*qt(0.975,5));La`
- `Lb<-c(pint1_b1-pint1_b2*qt(0.975,5),pint1_b1+pint1_b2*qt(0.975,5));Lb`

Podle počtu desetinných míst u směrodatných odchylek intervaly zaokrouhlíme

- `L1a<-round(La, digits=5);L1a`
- `L1b<-round(Lb, digits=4);L1b`

Zobrazené intervaly uvedeme do protokolu jako

`L(a)1,2 = <-0.00044;0.00028>`

`L(b)1,2 =< 0.1445;0.1548>`

Nyní vypočítáme limit detekce a limit stanovitelnosti

- `lod(linear3)`
- `> ld<-as.numeric(lod(linear3));ld`

`[1] 0.01262320 0.00181198`

- `> lq<-(ld*10/3);lq`

`[1] 0.042077325 0.006039934`

Mez detekce i stanovitelnosti se odečítá na ose x, takže nás v obou případech zajímá první hodnota, což je koncentrace, hodnoty si zapíšeme a poznamenejeme v protokolu.

(ld = mez detekce, lq = mez stanovitelnosti)

Dále zobrazíme elipsu pro parametry regresní přímky. Budeme potřebovat další knihovnu

- `library(ellipse)`

Zadáme funkci pro výpočet dat potřebných pro zobrazení elipsy

- `level = 0.975`
- `ell <- ellipse(linear3, which = c(1, 2), level = level)`

Zjištěné hodnoty vyneseme do grafu

- `plot(ell, type = "l", main = "Elipsa pro parametry", ylab = "Směrnice", xlab = "Úsek", xlim = c(-0.0005, 0.0004), ylim = c(0.143, 0.185))`

Pokud není elipsa vidět celá, změníme počáteční a koncové body os. Toho docílíme změnou čísel v závorce u `xlim=c(-0.12,0.25)` a `ylim=c(-0.02,0.04)`. Hodnoty v závorce udávají počátek a konec osy. Přidáme legendu

- `legend(-0.0005, 0.186, legend = c("elipsa", "intervaly spolehlivosti pro parametry"), col = c(1, 2), box.lwd = 0, box.col = "white", bg = "white", lty = c(1, 1), cex = 0.9, lwd = 1.5, pch = c("-", "-"))`

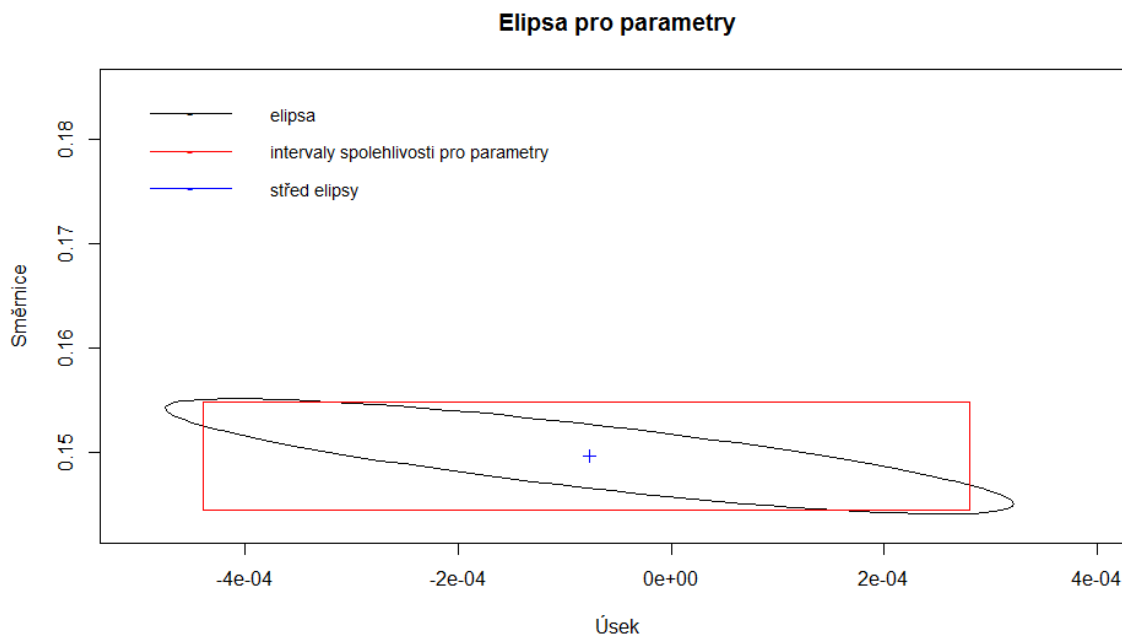
zobrazíme střed elipsy

- `points(linear3$coefficients[1], linear3$coefficients[2], pch = 3)`

a jednotlivé úsečky znázorňující intervaly spolehlivosti pro úsek a směrnici, ze kterých vznikne obdélník:

- `segments(L1a[1], L1b[1], L1a[2], L1b[1], col = 2)`
- `segments(L1a[1], L1b[2], L1a[2], L1b[2], col = 2)`
- `segments(L1a[1], L1b[1], L1a[1], L1b[2], col = 2)`
- `segments(L1a[2], L1b[2], L1a[2], L1b[1], col = 2)`

Graf uložíme a přiložíme do protokolu



Obrázek č. 11: Elipsa spolehlivosti pro parametry regresní přímky

Dále spustíme skript pro elipsy spolehlivosti a predikce. Celý skript označíme myší a spustíme. Tím vypočítáme všechna potřebná data pro vynesení elips do grafu:

- `level <- 0.95`
- `shape <- var(cbind(X, Y))`
- `center <- c(mean(konc), mean(prumer_blankless1))`
- `radiusC <- sqrt(2 * (length(konc)-1) * qf(level, 2, (length(konc) - 2)) / (length(konc) * (length(konc)-2)))`
- `radiusP <- radiusC * sqrt(length(konc)+1)`

- `ellcalc <- function(center, shape, radius, segments=1000){`
`segments=segments`
`angles <<- seq(from=0, to=(2*3.1415926), by=1/segments)`
`unit.circle <<- cbind(cos(angles), sin(angles))`
`ELLIPSE <<- t(center + radius * t(unit.circle %*% chol(shape)))`
`return(ELLIPSE)}`
- `ellC <- ellcalc(center, shape, radiusC)`
- `ellP <- ellcalc(center, shape, radiusP)`

Pokud je vše v pořádku, můžeme přistoupit k tvorbě grafu. Zadáme příkaz ke zpracování dat:

- `plot(konc, prumer_blankless, type="n", main="Elipsy", xlab="Koncentrace Cu (mg.l-1)", ylab="Absorbance (a.u.)", xlim=c(-0.12,0.25), ylim=c(-0.02,0.04))`

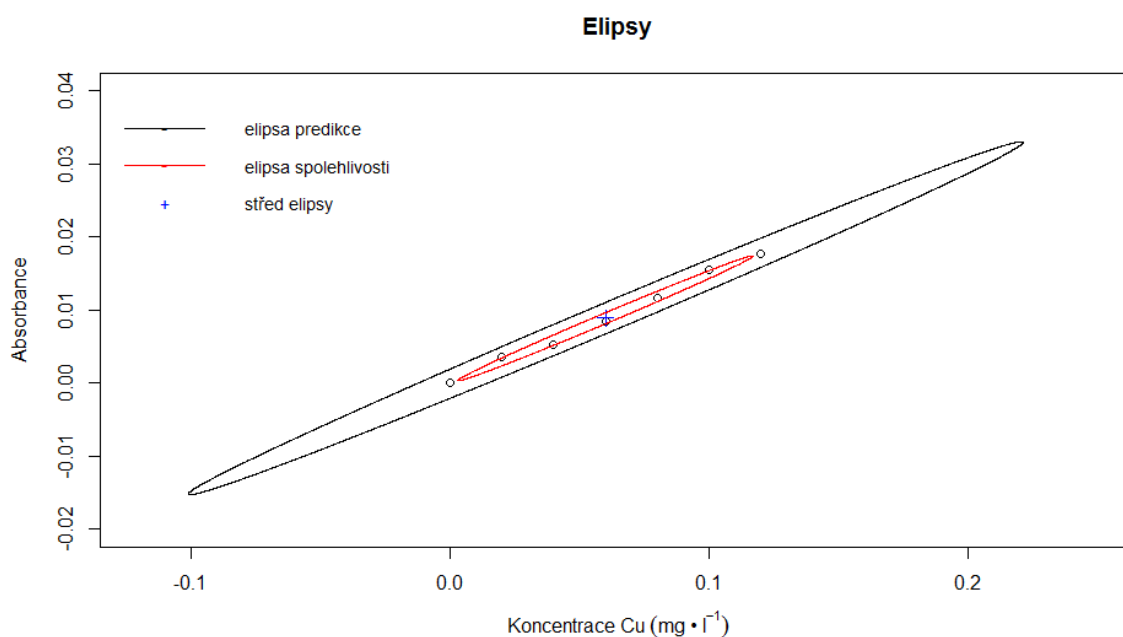
Dále zobrazíme legendu

- `legend(-0.13,0.04,c("elipsa predikce","elipsa spolehlivosti","průměr"), col=c(1,2,4), lty=c(1,1,0), box.lwd = 0, box.col = "white", bg = "white", cex=0.9, lwd=1.5, pch=c("-", "-", "+"))`

Pozici legendy opět můžeme upravit změnou čísel v závorce `legend(-0.12,0.04`. V grafu zobrazíme průměrné hodnoty bodů kalibrační přímky, střed elips a poté samotné elipsy:

- `points(konc, prumer_blankless,col=1)`
- `points(center[1], center[2], pch = 3,col=4,cex=1.25)`
- `points(ellC[,1],ellC[,2],col=2,type="l")`
- `points(ellP[,1],ellP[,2],col=1,type="l")`

Graf zobrazíme stejně jako předchozí, uložíme a přiložíme do protokolu.



Obrázek č. 11: Elipsa spolehlivosti a predikce

Nyní přikročíme k výpočtu koncentrace mědi ve vzorku. Načteme soubor, do kterého jsme uložili naměřené signály pro vzorky:

- `vzorky<-read.delim("c:\\kalibrace\\vzorky.txt",header=TRUE)`

Z každého sloupce vypočítáme průměr a v dalším kroku odečteme blank

- `prumer3<-apply(vzorky,2,mean);prumer3`

- `vzorky_blankless<-prumer3[1:length(prumer3)]-mean(blank);vzorky_blankless`

Zkontrolujeme zobrazené hodnoty. Dále vypočítáme koncentraci mědi, a to tak, že od průměrných hodnot signálu odečteme úsek a podělíme směrnicí

- `konc_Cu<-(((vzorky_blankless-pint1_a1)/pint1_b1)*5);konc_Cu`

Jednotlivé vypočtené koncentrace zobrazíme jako samostatné hodnoty opět pomocí funkce `as.matrix`

- `vzorky_Cu<-as.matrix(konc_Cu);vzorky_Cu`
- `vz1<-vzorky_Cu[1,1];vz1`
- `vz2<-vzorky_Cu[2,1];vz2`
- `vz3<-vzorky_Cu[3,1];vz3`

Poté vypočítáme intervaly spolehlivosti pro jednotlivé vzorky. Do výpočtu budeme potřebovat průměrnou hodnotu x a průměrnou hodnotu y z kalibrační přímky:

- `xpr<-mean(konc);xpr`
- `ypr<-mean(prumer_blankless1);ypr`

dále hodnoty y pro každou koncentraci vypočtené z regresní přímky

- `Y<-(pint1_a1+konc*pint1_b1);Y`

počet všech naměřených bodů použitých pro kalibraci

- `n<-(length(konc)*length(blank));n`

a také kolikrát byl změřen signál pro každý vzorek

- `m<-length(vzorky[,1]);m`

Pro výpočet $s_{y,x}$ potřebujeme znát sumu druhých mocnin z rozdílu každé naměřené hodnoty a hodnot vypočítaných pro jednotlivé koncentrace kalibračních roztoků. Musíme postupovat tak, že z původní matice naměřených hodnot po odečtení blanku vyčleníme jednotlivé řádky, od každého odečteme vektor vypočítaných hodnot, umocníme, nakonec sečteme. Označíme celý skript a spustíme

- `kp1<-kp_blankless[1,];kp1`
- `kp2<-kp_blankless[2,];kp2`
- `kp3<-kp_blankless[3,];kp3`
- `kp4<-kp_blankless[4,];kp4`
- `kp5<-kp_blankless[5,];kp5`
- `kp6<-kp_blankless[6,];kp6`
- `kp7<-kp_blankless[7,];kp7`
- `kpY1<-(sum((kp1-Y)^2));kpY1`

- $kpY2 \leftarrow -(\text{sum}((kp2-Y)^2)); kpY2$
- $kpY3 \leftarrow -(\text{sum}((kp3-Y)^2)); kpY3$
- $kpY4 \leftarrow -(\text{sum}((kp4-Y)^2)); kpY4$
- $kpY5 \leftarrow -(\text{sum}((kp5-Y)^2)); kpY5$
- $kpY6 \leftarrow -(\text{sum}((kp6-Y)^2)); kpY6$
- $kpY7 \leftarrow -(\text{sum}((kp7-Y)^2)); kpY7$
- $kpY \leftarrow -(kpY1+kpY2+kpY3+kpY4+kpY5+kpY6+kpY7); kpY$

Nyní můžeme dopočítat $s_{y,x}$

- $syx \leftarrow \text{sqrt}((kpY)/(n-2)); syx$

a poté směrodatné odchylky pro jednotlivé vzorky

- $s1 \leftarrow ((syx/pint1_b1) * \text{sqrt}((1/n)+(1/m)+(pint1_b2/(pint1_b1 * syx))^2 * (vz1-ypr)^2)); s1$
- $s2 \leftarrow ((syx/pint1_b1) * \text{sqrt}((1/n)+(1/m)+(pint1_b2/(pint1_b1 * syx))^2 * (vz2-ypr)^2)); s2$
- $s3 \leftarrow ((syx/pint1_b1) * \text{sqrt}((1/n)+(1/m)+(pint1_b2/(pint1_b1 * syx))^2 * (vz3-ypr)^2)); s3$

Ty zaokrouhlíme na dvě platné číslice a podle počtu míst za desetinnou tečkou pak zaokrouhlíme jednotlivé intervaly:

- $sx1 \leftarrow \text{signif}(s1, \text{digits}=2); sx1$
- $sx2 \leftarrow \text{signif}(s2, \text{digits}=2); sx2$
- $sx3 \leftarrow \text{signif}(s3, \text{digits}=2); sx3$

Nakonec vypočítáme intervaly spolehlivosti pro jednotlivé vzorky

- $Lvz1 \leftarrow -c(vz1-sx1 * qt(0.975,47), vz1+sx1 * qt(0.975,47)); Lvz1$
- $Lvz2 \leftarrow -c(vz2-sx2 * qt(0.975,47), vz2+sx2 * qt(0.975,47)); Lvz2$
- $Lvz3 \leftarrow -c(vz3-sx3 * qt(0.975,47), vz3+sx3 * qt(0.975,47)); Lvz3$

zaokrouhlíme na vhodný počet desetinných míst

- $Lvz1 \leftarrow \text{round}(Lv1, \text{digits}=3); Lvz1$
- $Lvz2 \leftarrow \text{round}(Lv2, \text{digits}=3); Lvz2$
- $Lvz3 \leftarrow \text{round}(Lv3, \text{digits}=3); Lvz3$

Intervaly spolehlivosti stanoveného množství Cu^{2+} si poznamenáme a uvedeme do protokolu jako

$$Lvz1 = \langle 0.127; 0.179 \rangle \text{ mg} \cdot \text{l}^{-1}$$

$$Lvz2 = \langle 0.146; 0.207 \rangle \text{ mg} \cdot \text{l}^{-1}$$

$$Lvz3 = \langle 0.152; 0.217 \rangle \text{ mg} \cdot \text{l}^{-1}$$

Stanovení mědi metodou přidavku standardu s doplněním na konstantní objem

Z výchozího roztoku o koncentraci Cu^{2+} $1 \text{ mg} \cdot \text{l}^{-1}$ připravíme roztok o koncentraci $0,4 \text{ mg} \cdot \text{l}^{-1}$. Z tohoto roztoku dále pipetujeme do 25 ml odměrných baněk postupně 0,00 ml; 1,25 ml; 2,50 ml; 3,75 ml; 5,00 ml; 6,25 ml a 7,50 ml. Do každé z odměrných baněk přidáme 5 ml nezředěného vína. Baňky doplníme po rysku 1% HNO_3 . Tímto získáme roztoky o koncentracích přidavku standardu mědi v $\text{mg} \cdot \text{l}^{-1}$: 0; 0,02; 0,04; 0,06; 0,08; 0,10; 0,12.

Jako blank použijeme roztok 1% HNO_3 . Každý roztok proměříme 7krát a hodnoty absorbance přepíšeme do Excelu na samostatný list a uložíme ve formátu txt pod názvem **prid_st** do složky kalibrace. **Dbáme na to, abychom místo desetinných čárek používali tečky.**

Je vhodné data přepsat do Excelu způsobem, který zobrazuje následující tabulka:

Tab. č. 6: Data získaná z měření pomocí metody přidavku standardu

blank	0 mg/l	0,02 mg/l	0,04 mg/l	0,06 mg/l	0,08 mg/l	0,10 mg/l	0,12 mg/l
0.003411	0.005832	0.008668	0.01149	0.01452	0.01686	0.02094	0.02291
0.00362	0.005789	0.008652	0.01144	0.01413	0.01687	0.02039	0.02302
0.003653	0.005916	0.009171	0.01123	0.01401	0.01656	0.02083	0.02407
0.003467	0.005946	0.008877	0.01131	0.01411	0.01730	0.02052	0.02334
0.003153	0.005714	0.009624	0.01149	0.01501	0.01696	0.02035	0.02358
0.003489	0.005509	0.009425	0.01159	0.01454	0.01766	0.02086	0.02309
0.003744	0.005717	0.009619	0.01182	0.01472	0.01725	0.02030	0.02306

Můžeme pokračovat se zpracováním v R Studiu. Pokud není v záložkách příkazového řádku otevřen soubor prid_st, klikneme na „File“, zvolíme „Open File“ a ze složky kalibrace vybereme soubor prid_st.R.

Myší modře označíme první řádek skriptu a klikneme na RUN, nebo zmáčkneme Ctrl+R, tím načteme vstupní data.

- `prid_st0<-read.delim("c:\\kalibrace\\prid_st.txt",header=TRUE)`

Pokud jsme vstupní data ukládali bez hlavičky, přepíšeme ve skriptu header=False. Opět umístění souboru (červený text) se může lišit. Kliknutím na název v okně prostředí zkontrolujeme načtená data.

Poté načteme vektor koncentrace

- `konc2<-c(0,0.02,0.04,0.06,0.08,0.1,0.12)`

Dále musíme ze vstupní matice vyčlenit blank, který je v prvním sloupci,

- `blank2<-prid_st0[,1];blank2`

a z něj vypočítat průměr

- `mean(blank2)`

dále blank z původní matice odstraníme:

- `prid_st1<-prid_st0[,-1];prid_st1`

Nyní můžeme od vstupní matice odečíst blank a dostaneme tak data ponížena o tuto hodnotu:

- `ps_blankless<-prid_st1[1:length(prid_st1)]-mean(blank2);ps_blankless`

Nakonec vypočítáme z každého sloupce průměr, který použijeme pro zobrazení průměrných hodnot v grafu kalibrační přímky:

- `prumer_blankless2<-apply(ps_blankless,2,mean);prumer_blankless2`

Nyní musíme data otestovat na odlehlost pomocí Grubbsova testu, ale testujeme data pro lineární závislost, takže napřed musíme tuto závislost vytvořit. V následujícím skriptu se vstupní data uspořádají do dvou sloupců a díky tomu budeme dále schopni udělat regresi pro všechna vstupní data. Označíme následující část skriptu a spustíme:

- `a<-length(ps_blankless[1,]);a`
- `b<-length(ps_blankless[,1]);b`
- `X<-rep(konc2,b);X`
- `Y<-ps_blankless[1,]`
- `for(i in 2:b){`
`YY<ps_blankless[i,]`
`Y<-as.numeric(c(Y,YY));Y`

Označíme další řádek, kterým R Studiu zadáme příkaz, že data jsou lineárního charakteru:

- `linear<-lm(Y~X);linear`

přivoláme potřebnou knihovnu

- `library(car)`

a provedeme samotné testování odlehlosti

- `outlierTest(linear,cutoff=0.05, n.max=10, order=TRUE)`

Pokud jsou data v pořádku, zobrazí se v okně konzoly:

No studentized residuals with Bonferroni p<0,05

Ale zobrazí se největší hodnota. Pokud zjistíme, že nějaká hodnota je odlehlá, poznamenáme si ji a uvedeme do protokolu.

Nyní provedeme testování úseku a směrnice dle nulových hypotéz $a = 0$ a $b = 1$, přivoláme potřebnou knihovnu a provedeme testování

- `library(lmtest)`
- `coefstest(linear, vcov. = NULL, df = NULL)`

Zobrazí se tabulka

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.0023536	0.0001966	11.972	7.171e-05	***
konc2	0.1447293	0.0027263	53.086	4.485e-08	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Pravděpodobnosti pro úsek i směrnici by se měly blížit nule. Pokud ne, měření zopakujeme. Hodnoty **úseku** („Intercept“) a **směrnice** („konc“) si poznamenejme a přepíšeme v legendě, aby odpovídaly kalibrační přímce pro naše data.

Dále zobrazíme graf kalibrační přímky

- `plot(X,Y,type="n",main="Kalibrační přímka",xlab="Koncentrace Cu (mg.l-1)",ylab="Absorbance (a.u.)",col=3)`
- `legend(-0.003,0.019,legend=c("kalibrační přímka - y = 0,0024 + 0,1447x", "chybové úsečky - int. spolehlivosti"),col=c(4,"red"),box.lwd = 0,box.col = "white",bg = "white",lty=c(1,1),cex=0.9,lwd=1.5,pch=c("-", "-"))`
- `abline(linear,col=4)`
- `points(konc2,prumer_blankless2,col=1)`

do grafu ale musíme přidat chybové úsečky, pro jejich výpočet potřebujeme směrodatnou odchylku z každého sloupce

- `sm.odch_blankless2<-apply(ps_blankless,2,sd);sm.odch_blankless2`

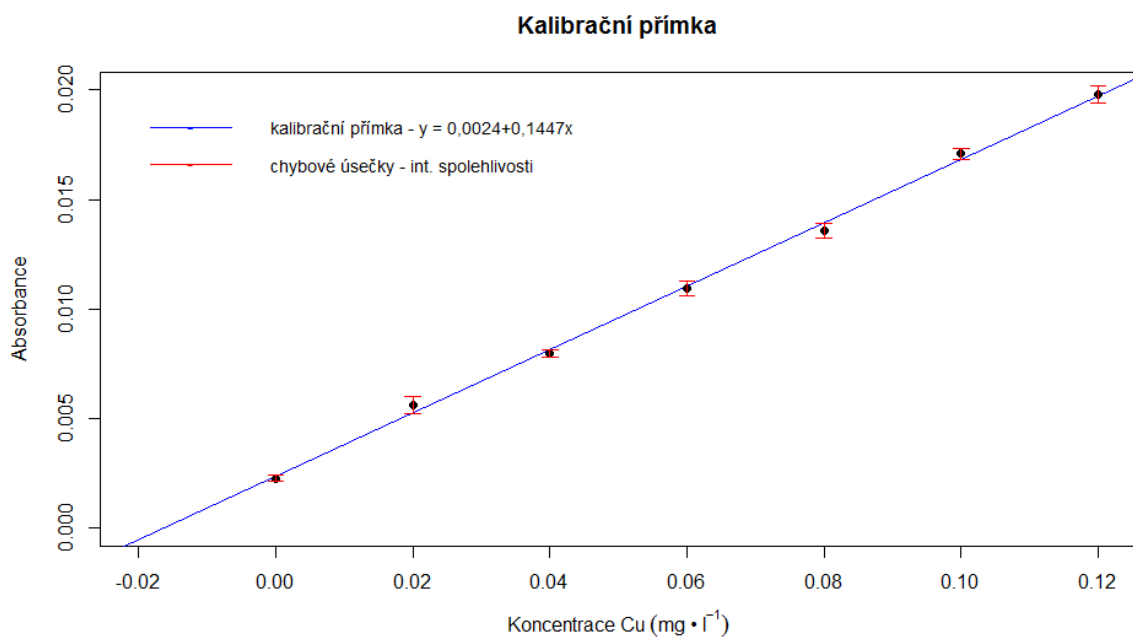
a hodnotu, kterou budeme přičítat a odečítat k průměrným hodnotám (směrodatná odchylka vynásobena kritickou hodnotou Studentova rozdělení pro $v = n - 1$, to celé poděleno odmocninou z n , což je počet měření pro každý kalibrační bod):

- `L2<-((sm.odch_blankless2*qt(0.975,6))/sqrt(length(blank2)));L2`

Přivoláme potřebnou knihovnu a zobrazíme chybové úsečky

- `library(Hmisc)`
- `errbar(konc2, prumer_blankless2, yplus=(prumer_blankless2+L2), yminus=(prumer_blankless2-L2), cap=0.015, add=T, errbar.col="red")`

Graf uložíme a přidáme do protokolu.

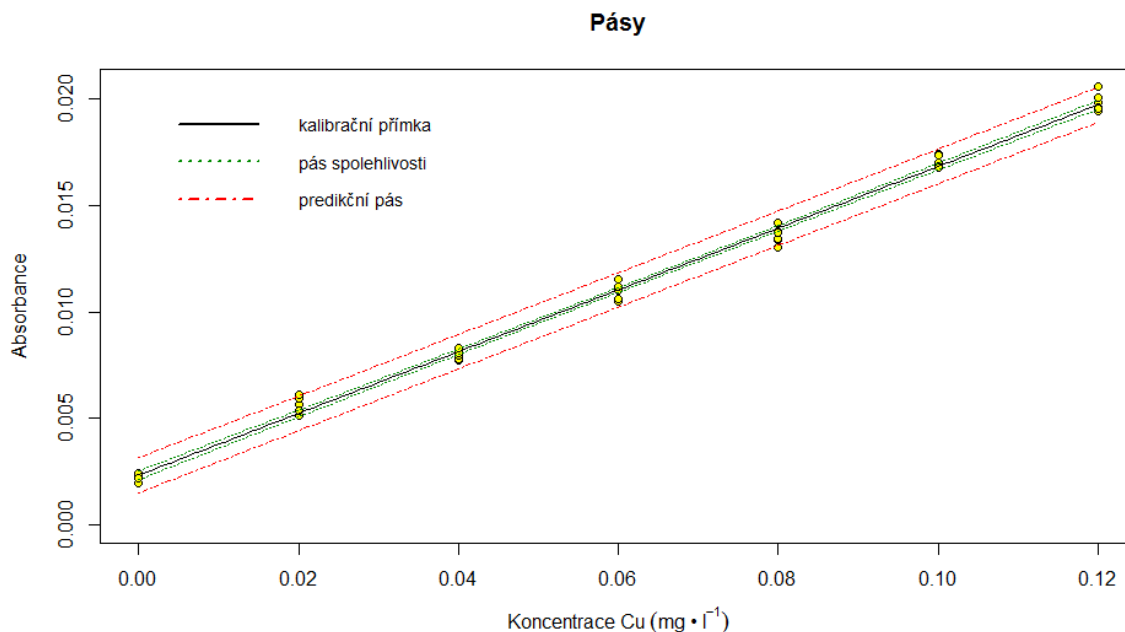


Obrázek č 14: Kalibrační přímka a chybové úsečky pro metodu přidavku

Nyní zobrazíme pásy spolehlivosti a predikce

- `library(chemCal)`
- `source("C:\\kalibrace\\calplot_primka.r")`
- `linear2<-calplot_primka(linear, xlim = c("auto", "auto"), ylim = c("auto", "auto"), xlab="Koncentrace Cu (mg.l-1)",ylab="Absorbance (a.u.)", alpha=0.05, varfunc = NULL)`

Graf uložíme a přiložíme do protokolu.



Obrázek č. 15: Pásky spolehlivosti a predikce

Poté vypočítáme mez detekce a stanovitelnosti, opět nás zajímají hodnoty na ose x, tudíž odečítáme první čísla, což odpovídá koncentraci.

- `lod(linear)`
- `> ld<-as.numeric(lod(linear));ld`

```
[1] 0.009460677 0.003722824
```

- `> lq<-/(ld*10/3);lq`

```
[1] 0.03153559 0.01240941
```

Hodnoty poznamenáme a uvedeme do protokolu.

Pokračujeme výpočtem intervalů spolehlivosti pro parametry. K tomu opět potřebujeme směrodatné odchylky úseku a směrnice

- `coefstest(linear, vcov. = NULL, df = NULL)`

které ale musíme zobrazit jako tabulku a jednotlivé hodnoty zobrazit samostatně

- `pint2<-as.matrix(coefstest(linear, vcov. = NULL, df = NULL));pint2`
- `pint2_a1<-pint2[1,1];pint2_a1`
- `pint2_a2<-pint2[1,2];pint2_a2`
- `pint2_b1<-pint2[2,1];pint2_b1`
- `pint2_b2<-pint2[2,2];pint2_b2`

Směrodatné odchylky zaokrouhlíme na dvě platné číslice a podle počtu desetinných míst zaokrouhlíme výsledné intervaly:

- `pa2<-signif(pint2_a2, digits=2);pa2`
- `pb2<-signif(pint2_b2, digits=2);pb2`

Dostaneme např.: `pa2 = 0.00010` a `pb2 = 0.0014`

Pokračujeme samotným výpočtem

- `La<-c(pint2_a1-pint2_a2*qt(0.975,5),pint2_a1+pint2_a2*qt(0.975,5));La`
- `Lb<-c(pint2_b1-pint2_b2*qt(0.975,5),pint2_b1+pint2_b2*qt(0.975,5));Lb`

Podle desetinných míst u směrodatných odchylek intervaly zaokrouhlíme

- `L2a<-round(La, digits=5);L1a`
- `L2b<-round(Lb, digits=4);L1b`

Zobrazené intervaly uvedeme do protokolu jako

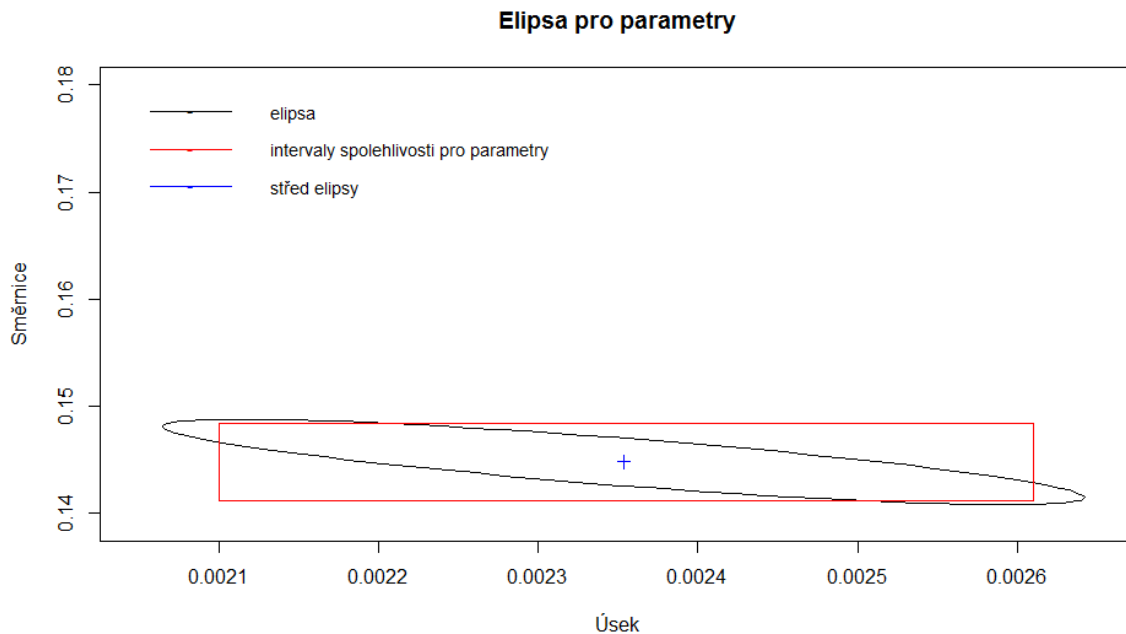
`L(a)1,2 = <0.00210;0.00261>`

`L(b)1,2 = <0.1411;0.1483>`

Poté zobrazíme elipsu pro parametry regresní přímky

- `library(ellipse)`
- `level = 0.975`
- `ell <- ellipse(linear, which = c(1, 2), level = level)`
- `plot(ell, type = "l", main="Elipsa pro parametry", ylab="Směrnice",
xlab="Úsek",xlim=c(0.00205,0.00265),ylim=c(0.137,0.165))`
- `points(linear$coefficients[1], linear$coefficients[2],pch=3)`
- `segments(L2a[1],L2b[1],L2a[2],L2b[1],col=2)`
- `segments(L2a[1],L2b[2],L2a[2],L2b[2],col=2)`
- `segments(L2a[1],L2b[1],L2a[1],L2b[2],col=2)`
- `segments(L2a[2],L2b[2],L2a[2],L2b[1],col=2)`
- `legend(0.00205,0.165,legend=c("elipsa", "intervaly spolehlivosti pro parametry"),
col=c(1,2),box.lwd = 0,box.col = "white",bg = "white", lty=c(1,1),
cex=0.9,lwd=1.5,pch=c("-", "-"))`

Pokud je elipsa příliš velká a není celá vidět, upravíme velikost os pomocí změny čísel v závorkách `xlim` a `ylim`, které udávají počátek a konec os. Stejným způsobem můžeme upravit pozici legendy. Graf uložíme a přidáme do protokolu



Obrázek č. 16: Elipsa pro parametry regresní přímky

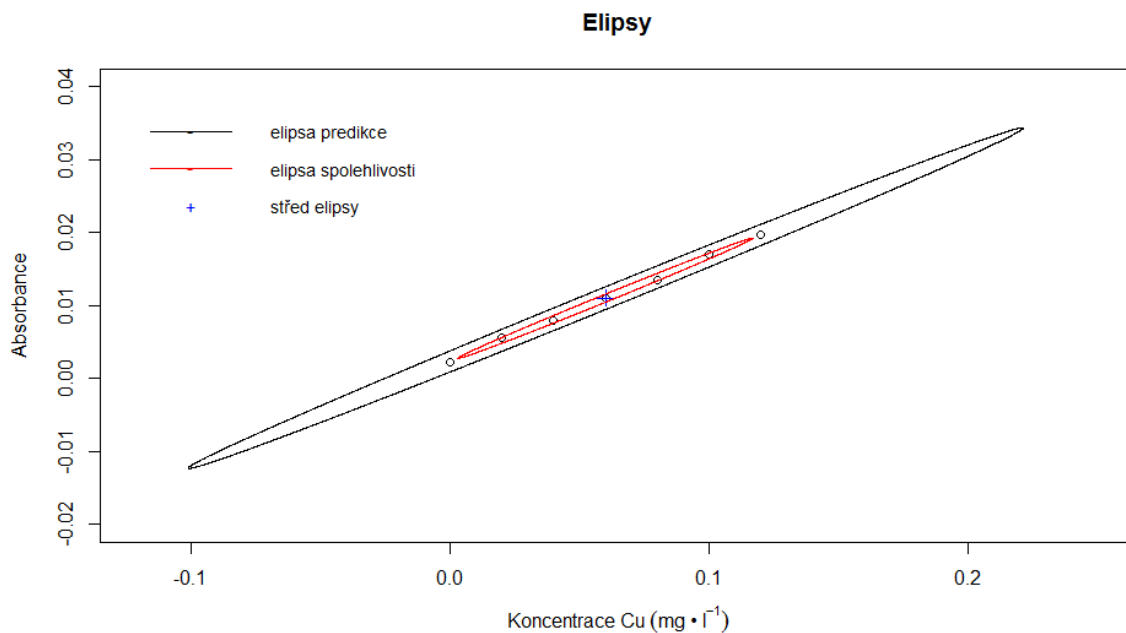
Nyní zobrazíme elipsy spolehlivosti a predikce

- `level <- 0.95`
- `shape <- var(cbind(X, Y))`
- `center <- c(mean(konc2), mean(prumer_blankless2))`
- `radiusC <- sqrt(2 * (length(konc2)-1) * qf(level, 2, (length(konc2) - 2)) / (length(konc2) * (length(konc2) - 2)))`
- `radiusP <- radiusC * sqrt(length(konc2) + 1)`

- `ellcalc <- function(center, shape, radius, segments=1000){`
`segments=segments`
`angles <- seq(from=0, to=(2*3.1415926), by=1/segments)`
`unit.circle <- cbind(cos(angles), sin(angles))`
`ELLIPSE <- t(center + radius * t(unit.circle %*% chol(shape)))`
`return(ELLIPSE)}`
- `ellC <- ellcalc(center, shape, radiusC)`
- `ellP <- ellcalc(center, shape, radiusP)`

- `plot(X, Y, type="p", main="elipsy", xlab="Koncentrace Cu (mg.l-1)", ylab="Absorbance (a.u.)", xlim=c(-0.12,0.25), ylim=c(-0.02,0.04))`
- `legend(-0.12,0.039,c("elipsa predikce","elipsa spolehlivosti","průměr"), col=c(1,2,4), lty=c(1,1,0), box.lwd = 0, box.col = "white", bg = "white", cex=0.9, lwd=1.5, pch=c("-", "-", "+"))`
- `points(konc2,prumer_blankless2)`
- `points(center[1], center[2], pch = 3,col=4,cex=1.25)`
- `points(ellC[,1],ellC[,2],col=2,type="l")`
- `points(ellP[,1],ellP[,2],col=1,type="l")`

Obrázek uložíme a přidáme k protokolu.



Obrázek č. 16: Elipsa spolehlivosti a elipsa predikce

Nyní můžeme přikročit k výpočtu koncentrace mědi ve vzorku

- `konc.Cu<-((pint2_a1/pint2_b1)*5);konc.Cu`

Pro výpočet intervalu spolehlivosti potřebujeme směrodatnou odchylku, pro její výpočet zase potřebujeme směrodatnou odchylku $s_{y,x}$:

- `xpr2<-mean(konc2);xpr2`
- `ypr2<-mean(prumer_blankless2);ypr2`
- `Q2<-sum(7*(konc2-xpr2)^2);Q2`
- `Y2<-((pint2_a1+konc2*pint2_b1);Y2`

- $n <- (\text{length}(\text{konc2}) * \text{length}(\text{blank2})); n$
- $ps1 <- ps_blankless[1,]; ps1$
- $ps2 <- ps_blankless[2,]; ps2$
- $ps3 <- ps_blankless[3,]; ps3$
- $ps4 <- ps_blankless[4,]; ps4$
- $ps5 <- ps_blankless[5,]; ps5$
- $ps6 <- ps_blankless[6,]; ps6$
- $ps7 <- ps_blankless[7,]; ps7$
- $psY1 <- (\text{sum}((ps1 - Y2)^2)); psY1$
- $psY2 <- (\text{sum}((ps2 - Y2)^2)); psY2$
- $psY3 <- (\text{sum}((ps3 - Y2)^2)); psY3$
- $psY4 <- (\text{sum}((ps4 - Y2)^2)); psY4$
- $psY5 <- (\text{sum}((ps5 - Y2)^2)); psY5$
- $psY6 <- (\text{sum}((ps6 - Y2)^2)); psY6$
- $psY7 <- (\text{sum}((ps7 - Y2)^2)); psY7$
- $psY <- (psY1 + psY2 + psY3 + psY4 + psY5 + psY6 + psY7); psY$
- $s_{yx2} <- \text{sqrt}(psY / (n - 2)); s_{yx2}$

Nyní můžeme vypočítat směrodatnou odchylku s_x

- $sx <- ((s_{yx2} / \text{pint2_b1}) * \text{sqrt}((1/n) + ((0 - ypr2)^2 / (\text{pint2_b1}^2 * Q2))))); sx$

Směrodatnou odchylku zaokrouhlíme na dvě platné číslice a podle počtu míst za desetinnou tečkou pak také zaokrouhlíme interval spolehlivosti

- $sx1 <- \text{signif}(sx, \text{digits} = 2); sx1$

Vypočítáme samotný interval spolehlivosti

- $L_{vz} <- c(\text{konc.Cu} - qt(0.975, 47) * sx1, \text{konc.Cu} + qt(0.975, 47) * sx1); L_{vz}$

Interval zaokrouhlíme

- $L_{vz} <- \text{round}(L_v, \text{digits} = 5); L_{vz}$

a stanovený obsah Cu^{2+} uvedeme do protokolu jako

$L_{vz} = \langle 0.07962; 0.08300 \rangle \text{ mg} \cdot \text{l}^{-1}$.