Exploratory Data Analysis

Fabian Peri

October 5, 2018

```
# packages needed for chapter 1
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(vioplot)
```

```
## Loading required package: sm
```

```
## Package 'sm', version 2.2-5.6: type help(sm) for summary information
```

```
library(ascii)
```

```
##
## Attaching package: 'ascii'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(descr)
```

```
# Import the datasets needed for chapter 1
PSDS_PATH <- file.path('C:/Users/fabia/Desktop', 'psds_data')
dir.create(file.path(PSDS_PATH, 'figures'))
```

```
## Warning in dir.create(file.path(PSDS_PATH, "figures")): 'C:\Users\fabia
## \Desktop\psds_data\figures' already exists
```

```
state <- read.csv(file.path(PSDS_PATH, 'data', 'state.csv'))
dfw <- read.csv(file.path(PSDS_PATH, 'data', 'dfw_airline.csv'))
sp500_px <- read.csv(file.path(PSDS_PATH, 'data', 'sp500_px.csv'))
sp500_sym <- read.csv(file.path(PSDS_PATH, 'data', 'sp500_sym.csv'), stringsAsFactors = FALSE)
kc_tax <- read.csv(file.path(PSDS_PATH, 'data', 'kc_tax.csv'))
lc_loans <- read.csv(file.path(PSDS_PATH, 'data', 'lc_loans.csv'))
airline_stats <- read.csv(file.path(PSDS_PATH, 'data', 'airline_stats.csv'), stringsAsFactors = FALSE)
airline_stats$airline <- ordered(airline_stats$airline, levels=c('Alaska', 'American', 'Jet Blue', 'Del
```

```
## Code to create state table
state_asc <- state

state_asc[["Population"]] <- formatC(state_asc[["Population"]], format="d", digits=0, big.mark=",")

ascii(state_asc[1:8,], digits=c(0, 0,1), align=c("l", "l", "r", "r"), caption="A few rows of the +data.
```

```
## Warning in rep(rownames, length = nrow(x)): 'x' is NULL so the result will
## be NULL
```

```
## Warning in rep(colnames, length = ncol(x)): 'x' is NULL so the result will
## be NULL
```

```
## .A few rows of the +data.frame state+ of population and murder rate by state.
## |=======================================================================
## 1.1+| <h| State       >h| Population >h| Murder.Rate <h| Abbreviation
## <| 1 <| Alabama     >| 4,779,736  >| 5.7         <| AL
## <| 2 <| Alaska      >| 710,231    >| 5.6         <| AK
## <| 3 <| Arizona     >| 6,392,017  >| 4.7         <| AZ
## <| 4 <| Arkansas    >| 2,915,918  >| 5.6         <| AR
## <| 5 <| California  >| 37,253,956 >| 4.4         <| CA
## <| 6 <| Colorado    >| 5,029,196  >| 2.8         <| CO
## <| 7 <| Connecticut >| 3,574,097  >| 2.4         <| CT
## <| 8 <| Delaware    >| 897,934    >| 5.8         <| DE
## |=======================================================================
```

The basic metric for location is the mean, but it can be sensitive to extreme values (outlier). Other metrics (median, trimmed mean) are more robust.

```
## Code snippet 1.1
mean(state[["Population"]])
```

```
## [1] 6162876
```

```
mean(state[["Population"]], trim=0.1)
```

```
## [1] 4783697
```

```
median(state[["Population"]])
```

```
## [1] 4436370
```

```
## Code snippet 1.2
mean(state[["Murder.Rate"]])
```

```
## [1] 4.066
```

```
library("matrixStats")
```

```
##
## Attaching package: 'matrixStats'
```

```
## The following object is masked from 'package:dplyr':
##
##     count
```

```
weighted.mean(state[["Murder.Rate"]], w=state[["Population"]])
```

```
## [1] 4.445834
```

```
weightedMedian(state[["Murder.Rate"]], w=state[["Population"]])
```

```
## [1] 4.4
```

```
## Code snippet 1.3
sd(state[["Population"]])
```

```
## [1] 6848235
```

```
IQR(state[["Population"]])
```

```
## [1] 4847308
```

```
mad(state[["Population"]])
```

```
## [1] 3849870
```

```
## Code snippet 1.4
quantile(state[["Murder.Rate"]], p=c(.05, .25, .5, .75, .95))
```

```
##    5%   25%   50%   75%   95%
## 1.600 2.425 4.000 5.550 6.510
```

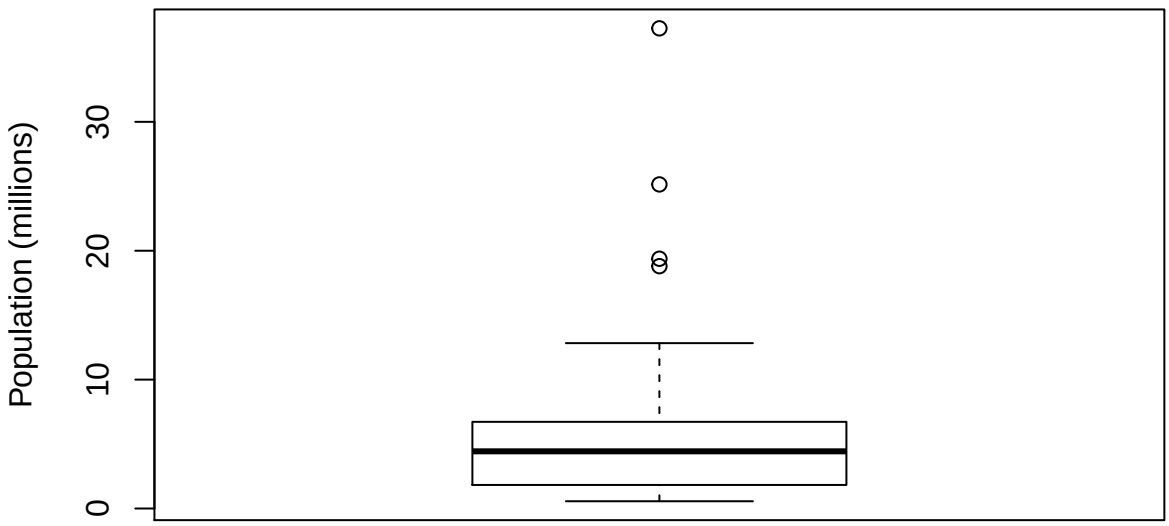
```
## Code to create PercentileTable
ascii(
  quantile(state[["Murder.Rate"]], p=c(.05, .25, .5, .75, .95)),
  include.rownames=FALSE, include.colnames=TRUE, digits=rep(2,5), align=rep("r", 5),
  caption="Percentiles of murder rate by state.")
```

```
## Warning in rep(rownames, length = nrow(x)): 'x' is NULL so the result will
## be NULL
```

```
## Warning in rep(colnames, length = ncol(x)): 'x' is NULL so the result will
## be NULL
```

```
## .Percentiles of murder rate by state.
## |=======================================
## >| 5%   >| 25%  >| 50%  >| 75%  >| 95%
## >| 1.60 >| 2.42 >| 4.00 >| 5.55 >| 6.51
## |=======================================
```

```
## Code snippet 1.5
boxplot(state[["Population"]]/1000000, ylab="Population (millions)")
```




```
## Code for Figure 2
png(filename=file.path(PSDS_PATH, "figures", "psds_0102.png"), width = 3, height=4, units='in', res=300)
par(mar=c(0,4,0,0)+.1)
boxplot(state[["Population"]]/1000000, ylab="Population (millions)")
dev.off()
```

```
## pdf
##   2
```

```
## Code snippet 1.6
breaks <- seq(from=min(state[["Population"]]), to=max(state[["Population"]]), length=11)
pop_freq <- cut(state[["Population"]], breaks=breaks, right=TRUE, include.lowest = TRUE)
state['PopFreq'] <- pop_freq
table(pop_freq)
```

```
## pop_freq
## [5.64e+05,4.23e+06]  (4.23e+06,7.9e+06]  (7.9e+06,1.16e+07]
##                  24                   14                   6
## (1.16e+07,1.52e+07] (1.52e+07,1.89e+07] (1.89e+07,2.26e+07]
##                   2                    1                   1
## (2.26e+07,2.62e+07] (2.62e+07,2.99e+07] (2.99e+07,3.36e+07]
##                   1                    0                   0
## (3.36e+07,3.73e+07]
##                   1
```

```
## Code for FreqTable
state_abb <- state %>%
  arrange(Population) %>%
```

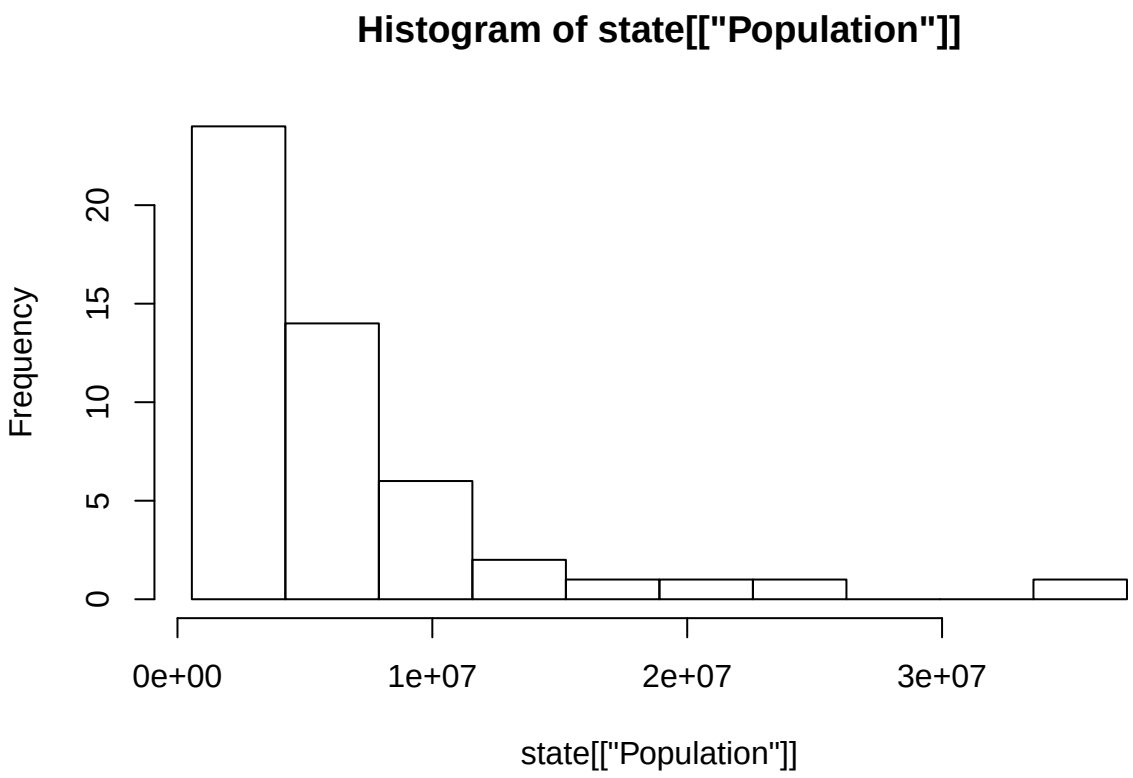
```
  group_by(PopFreq) %>%
  summarize(state = paste(Abbreviation, collapse=","), .drop=FALSE) %>%
  complete(PopFreq, fill=list(state='')) %>%
  select(state)

state_abb <- unlist(state_abb)

lower_br <- formatC(breaks[1:10], format="d", digits=0, big.mark=",")
upper_br <- formatC(c(breaks[2:10]-1, breaks[11]), format="d", digits=0, big.mark=",")

pop_table <- data.frame("BinNumber"=1:10,
                        "BinRange"=paste(lower_br, upper_br, sep="-"),
                        "Count"=as.numeric(table(pop_freq)),
                        "States"=state_abb)
ascii(pop_table, include.rownames=FALSE, digits=c(0, 0, 0, 0), align=c("l", "r", "r", "l"),
      caption="A frequency table of population by state.")
```

```
## Warning in rep(rownames, length = nrow(x)): 'x' is NULL so the result will
## be NULL
```

```
## Warning in rep(colnames, length = ncol(x)): 'x' is NULL so the result will
## be NULL
```

```
## .A frequency table of population by state.
## |=============================================================================================================
## <h| BinNumber >h| BinRange              >h| Count <h| States
## <| 1         >| 563,626-4,232,658     >| 24    <| WY,VT,ND,AK,SD,DE,MT,RI,NH,ME,HI,ID,NE,WV,NM,NV,UT
## <| 2         >| 4,232,659-7,901,691   >| 14    <| KY,LA,SC,AL,CO,MN,WI,MD,MO,TN,AZ,IN,MA,WA
## <| 3         >| 7,901,692-11,570,724  >| 6     <| VA,NJ,NC,GA,MI,OH
## <| 4         >| 11,570,725-15,239,757 >| 2     <| PA,IL
## <| 5         >| 15,239,758-18,908,790 >| 1     <| FL
## <| 6         >| 18,908,791-22,577,823 >| 1     <| NY
## <| 7         >| 22,577,824-26,246,856 >| 1     <| TX
## <| 8         >| 26,246,857-29,915,889 >| 0     <|
## <| 9         >| 29,915,890-33,584,922 >| 0     <|
## <| 10        >| 33,584,923-37,253,956 >| 1     <| CA
## |=============================================================================================================
```

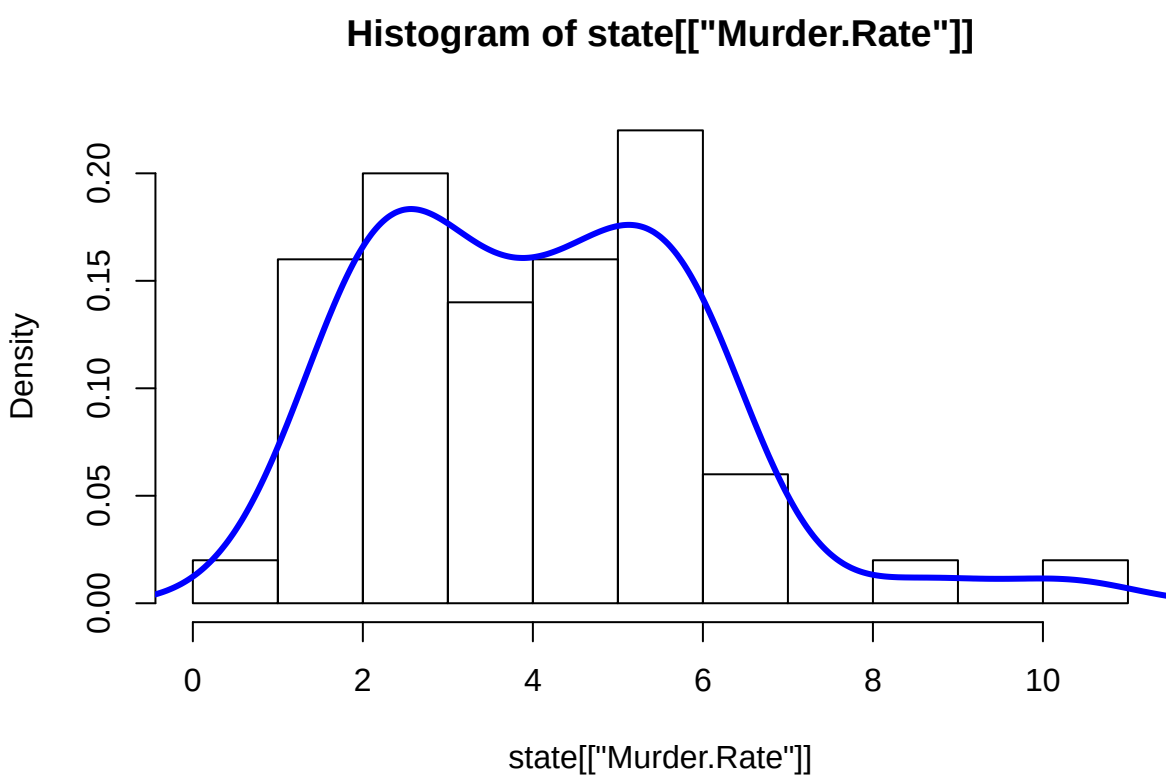
```
## Code snippet 1.7
hist(state[["Population"]], breaks=breaks)
```




```
## Code for Figure 3
png(filename=file.path(PSDS_PATH, "figures", "psds_0103.png"),  width = 4, height=4, units='in', res=300
par(mar=c(4,4,0,0)+.1)
pop_hist <- hist(state[["Population"]], breaks=breaks,
                 xlab="Population", main="")
dev.off()
```

```
## pdf
##   2
```

```
## Code snippet 1.8
hist(state[["Murder.Rate"]], freq=FALSE )
lines(density(state[["Murder.Rate"]]), lwd=3, col="blue")
```




```
## Code for Figure 4
png(filename=file.path(PSDS_PATH, "figures", "psds_0104.png"),  width = 4, height=4, units='in', res=300
par(mar=c(4,4,0,0)+.1)
hist(state[["Murder.Rate"]], freq=FALSE, xlab="Murder Rate (per 100,000)", main="" )
lines(density(state[["Murder.Rate"]]), lwd=3, col="blue")
dev.off()
```

```
## pdf
##   2
```

```
## Code for AirportDelays
ascii(
  100*as.matrix(dfw/sum(dfw)),
  include.rownames=FALSE, include.colnames=TRUE, digits=rep(2,5), align=rep("r", 5),
  caption="Percentage of delays by cause at Dallas-Ft. Worth airport.")
```

```
## Warning in rep(rownames, length = nrow(x)): 'x' is NULL so the result will
## be NULL
```

```
## Warning in rep(colnames, length = ncol(x)): 'x' is NULL so the result will
## be NULL
```

```
## .Percentage of delays by cause at Dallas-Ft. Worth airport.
## |=======================================================
## >| Carrier >| ATC   >| Weather >| Security >| Inbound
## >| 23.02   >| 30.40 >| 4.03    >| 0.12     >| 42.43
## |=======================================================
```

```
## Code for figure 5
png(filename=file.path(PSDS_PATH, "figures", "psds_0105.png"),  width = 4, height=4, units='in', res=300
par(mar=c(4, 4, 0, 1) + .1)
barplot(as.matrix(dfw)/6, cex.axis = 0.8, cex.names = 0.7)
dev.off()
```

```
## pdf
##   2
```

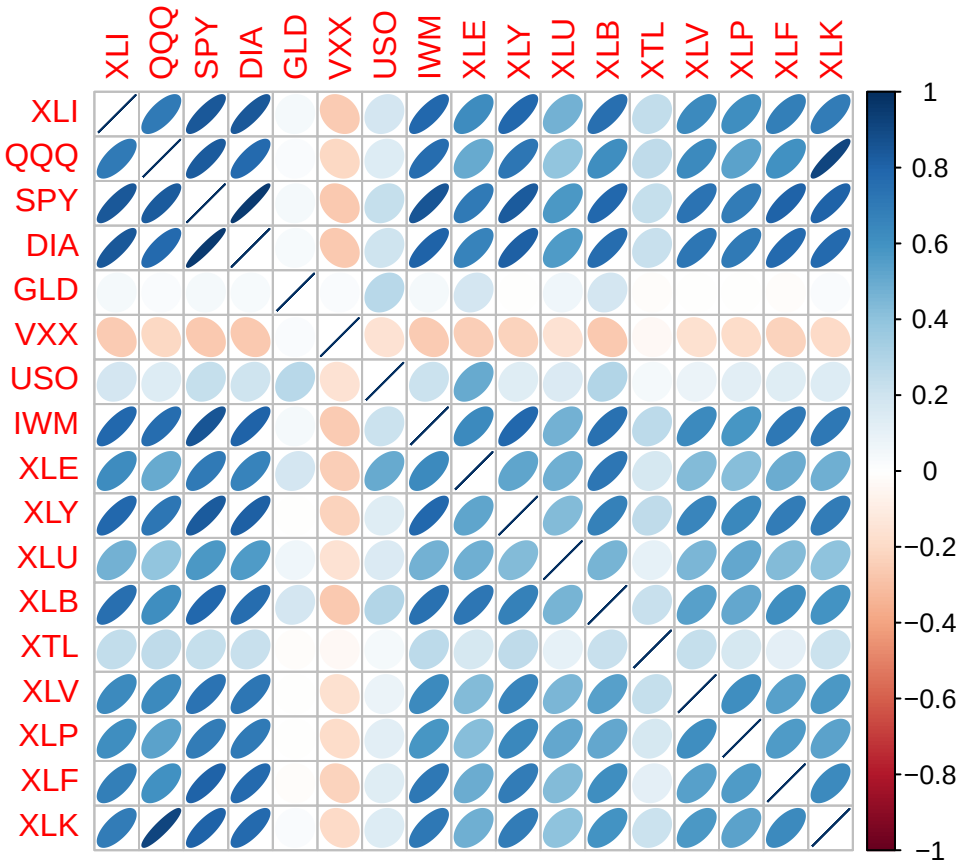
```
## Code for CorrTable (Table 1.7)
telecom <- sp500_px[, sp500_sym[sp500_sym$sector=="telecommunications_services", 'symbol']]
telecom <- telecom[row.names(telecom)>"2012-07-01", ]
telecom_cor <- cor(telecom)
ascii(telecom_cor, digits=c( 3,3,3,3,3), align=c("l", "r", "r", "r", "r", "r"), caption="Correlation be
      include.rownames = TRUE, include.colnames = TRUE)
```

```
## Warning in rep(rownames, length = nrow(x)): 'x' is NULL so the result will
## be NULL
```

```
## Warning in rep(colnames, length = ncol(x)): 'x' is NULL so the result will
## be NULL
```

```
## .Correlation between telecommunication stock returns.
## |=====================================================
## 1.1+| >| T     >| CTL   >| FTR   >| VZ    >| LVLT
## <| T    >| 1.000 >| 0.455 >| 0.359 >| 0.681 >| 0.082
## <| CTL  >| 0.455 >| 1.000 >| 0.435 >| 0.448 >| 0.096
## <| FTR  >| 0.359 >| 0.435 >| 1.000 >| 0.349 >| 0.111
## <| VZ   >| 0.681 >| 0.448 >| 0.349 >| 1.000 >| 0.096
## <| LVLT >| 0.082 >| 0.096 >| 0.111 >| 0.096 >| 1.000
## |=====================================================
```

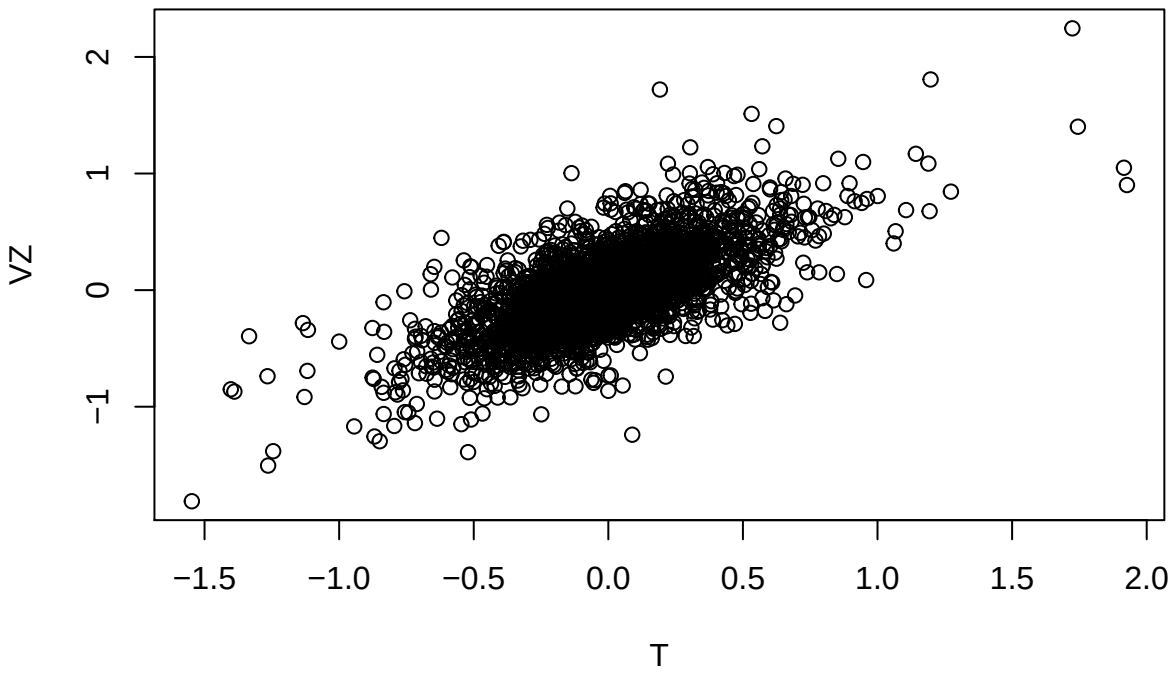
```
## Code snippet 1.10
etfs <- sp500_px[row.names(sp500_px)>"2012-07-01",
                 sp500_sym[sp500_sym$sector=="etf", 'symbol']]
corrplot(cor(etfs), method = "ellipse")
```




```
## Code for figure 6
png(filename=file.path(PSDS_PATH, "figures", "psds_0106.png"), width = 4, height=4, units='in', res=300)
etfs <- sp500_px[row.names(sp500_px)>"2012-07-01", sp500_sym[sp500_sym$sector=="etf", 'symbol']]
library(corrplot)
corrplot(cor(etfs), method = "ellipse")
dev.off()
```

```
## pdf
##   2
```

```
## Code snippet 1.11
plot(telecom$T, telecom$VZ, xlab="T", ylab="VZ")
```



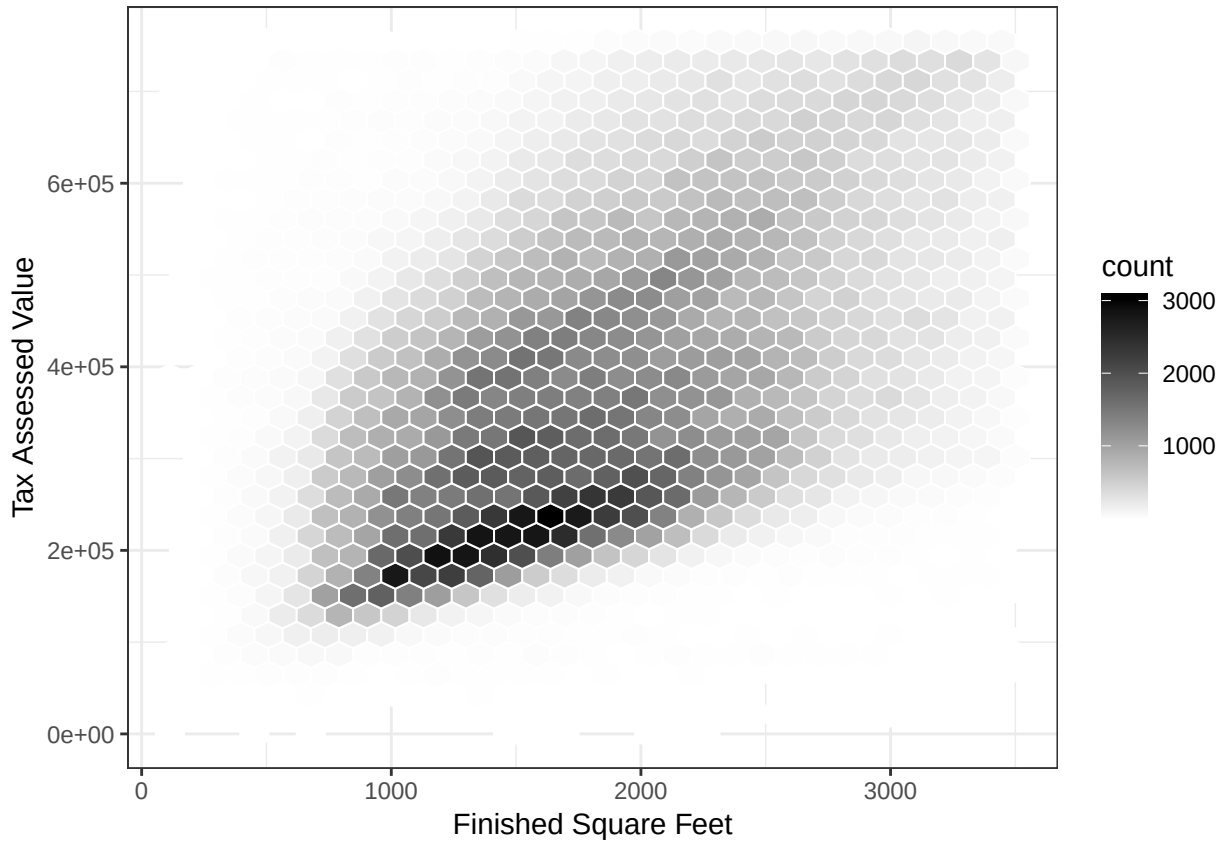

```
## Code for Figure 7
png(filename=file.path(PSDS_PATH, "figures", "psds_0107.png"),  width = 4, height=4, units='in', res=300
par(mar=c(4,4,0,1)+.1)
plot(telecom$T, telecom$VZ, xlab="T", ylab="VZ", cex=.8)
abline(h=0, v=0, col="grey")
dev.off()
```

```
## pdf
##   2
```

```
## Code snippet 1.12
kc_tax0 <- subset(kc_tax, TaxAssessedValue < 750000 & SqFtTotLiving>100 &
                  SqFtTotLiving<3500)
nrow(kc_tax0)
```

```
## [1] 432693
```

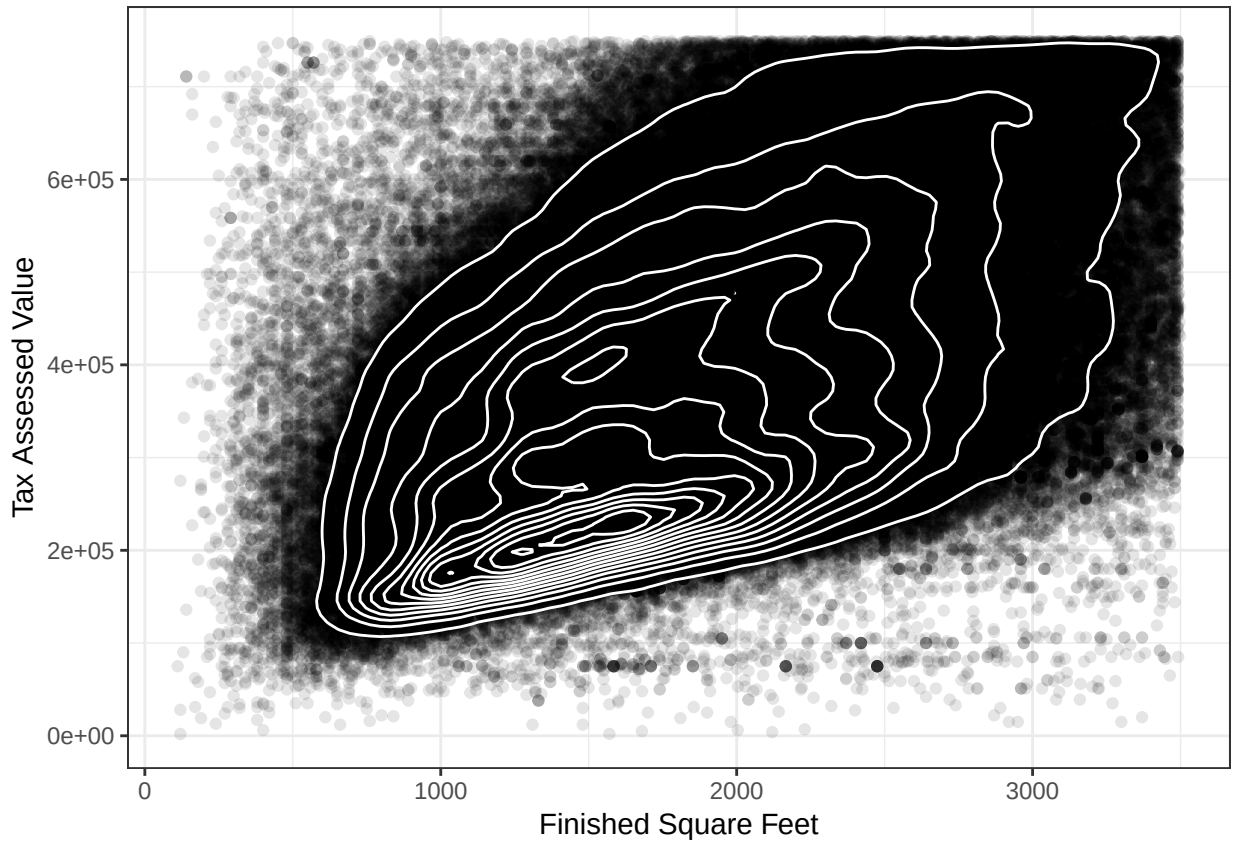
```
## Code snippet 1.13
ggplot(kc_tax0, (aes(x=SqFtTotLiving, y=TaxAssessedValue))) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient(low="white", high="black") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
```




```
## Code for figure 8
png(filename=file.path(PSDS_PATH, "figures", "psds_0108.png"),  width = 4, height=4, units='in', res=300
ggplot(kc_tax0, (aes(x=SqFtTotLiving, y=TaxAssessedValue))) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient(low="white", high="black") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
dev.off()
```

```
## pdf
##   2
```

```
## Code snippet 1.14
ggplot(kc_tax0, aes(SqFtTotLiving, TaxAssessedValue)) +
  theme_bw() +
  geom_point( alpha=0.1) +
  geom_density2d(colour="white") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
```




```
## Code for figure 9

png(filename=file.path(PSDS_PATH, "figures", "psds_0109.png"),  width = 4, height=4, units='in', res=300
ggplot(kc_tax0, aes(SqFtTotLiving, TaxAssessedValue)) +
  theme_bw() +
  geom_point(colour="blue", alpha=0.1) +
  geom_density2d(colour="white") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
dev.off()
```

```
## pdf
##   2
```

```
## Code snippet 1.15

## Code for CrossTabs
x_tab <- CrossTable(lc_loans$grade, lc_loans$status,
                    prop.c=FALSE, prop.chisq=FALSE, prop.t=FALSE)

tots <- cbind(row.names(x_tab$tab), format(cbind(x_tab$tab, x_tab$rs)))
props <- cbind("", format(cbind(x_tab$prop.row, x_tab$rs/x_tab$gt), digits=1))
c_tot <- c("Total", format(c(x_tab$cs, x_tab$gt)))

asc_tab <- matrix(nrow=nrow(tots)*2+1, ncol=ncol(tots))
colnames(asc_tab) <- c("Grade", colnames(x_tab$tab), "Total")
idx <- seq(1, nrow(asc_tab)-1, by=2)
asc_tab[idx,] <- tots
```

```r
asc_tab[idx+1,] <- props
asc_tab[nrow(asc_tab), ] <- c_tot

ascii(asc_tab,  align=c("l", "r", "r", "r", "r"), include.rownames = FALSE, include.colnames = TRUE)
```

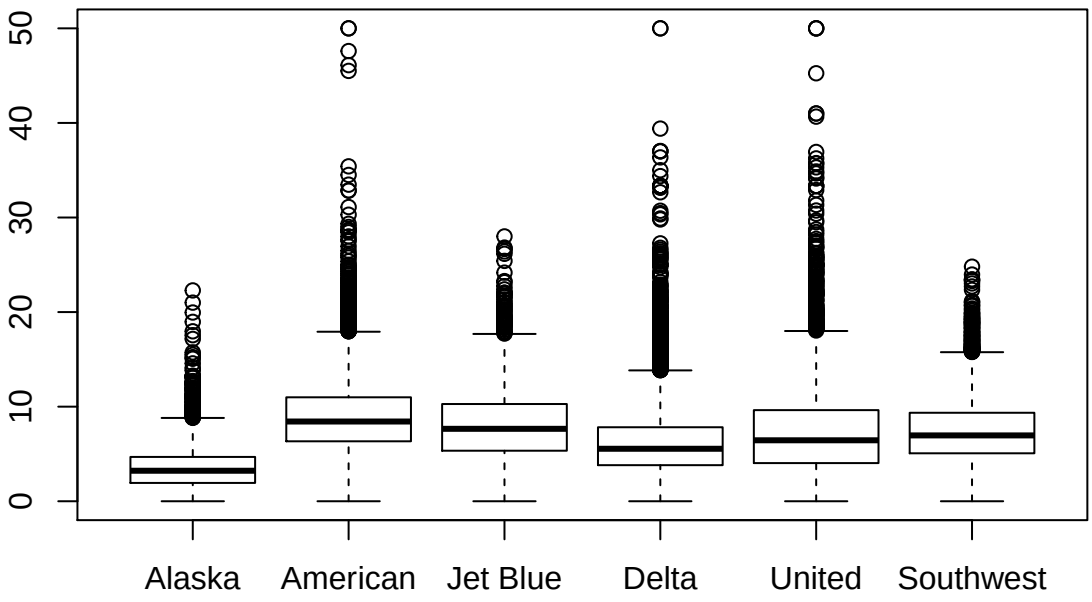
```
## Warning in rep(rownames, length = nrow(x)): 'x' is NULL so the result will
## be NULL
```

```
## Warning in rep(colnames, length = ncol(x)): 'x' is NULL so the result will
## be NULL
```

```
## |=====================================================================
## <| Grade >| Charged Off >| Current >| Fully Paid >| Late  <| Total
## <| 0.2   >| 146         >| 661     >| 216        >| 63    <| 1086
## <|       >| 1e-01       >| 6e-01   >| 2e-01      >| 6e-02 <| 2e-03
## <| 0.4   >| 96          >| 524     >| 138        >| 42    <| 800
## <|       >| 1e-01       >| 7e-01   >| 2e-01      >| 5e-02 <| 2e-03
## <| 0.6   >| 63          >| 360     >| 101        >| 45    <| 569
## <|       >| 1e-01       >| 6e-01   >| 2e-01      >| 8e-02 <| 1e-03
## <| 0.8   >| 51          >| 228     >| 100        >| 32    <| 411
## <|       >| 1e-01       >| 6e-01   >| 2e-01      >| 8e-02 <| 9e-04
## <| 1     >| 53          >| 217     >| 88         >| 17    <| 375
## <|       >| 1e-01       >| 6e-01   >| 2e-01      >| 5e-02 <| 8e-04
## <| 1.2   >| 400         >| 2598    >| 686        >| 161   <| 3845
## <|       >| 1e-01       >| 7e-01   >| 2e-01      >| 4e-02 <| 9e-03
## <| 1.4   >| 333         >| 1875    >| 576        >| 140   <| 2924
## <|       >| 1e-01       >| 6e-01   >| 2e-01      >| 5e-02 <| 6e-03
## <| 1.6   >| 302         >| 1748    >| 448        >| 134   <| 2632
## <|       >| 1e-01       >| 7e-01   >| 2e-01      >| 5e-02 <| 6e-03
## <| 1.8   >| 267         >| 1309    >| 347        >| 91    <| 2014
## <|       >| 1e-01       >| 6e-01   >| 2e-01      >| 5e-02 <| 4e-03
## <| 2     >| 224         >| 914     >| 271        >| 80    <| 1489
## <|       >| 2e-01       >| 6e-01   >| 2e-01      >| 5e-02 <| 3e-03
## <| 2.2   >| 597         >| 6401    >| 1435       >| 299   <| 8732
## <|       >| 7e-02       >| 7e-01   >| 2e-01      >| 3e-02 <| 2e-02
## <| 2.4   >| 670         >| 6011    >| 1470       >| 301   <| 8452
## <|       >| 8e-02       >| 7e-01   >| 2e-01      >| 4e-02 <| 2e-02
## <| 2.6   >| 561         >| 4798    >| 1186       >| 256   <| 6801
## <|       >| 8e-02       >| 7e-01   >| 2e-01      >| 4e-02 <| 2e-02
## <| 2.8   >| 547         >| 4066    >| 1001       >| 251   <| 5865
## <|       >| 9e-02       >| 7e-01   >| 2e-01      >| 4e-02 <| 1e-02
## <| 3     >| 467         >| 3363    >| 857        >| 267   <| 4954
## <|       >| 9e-02       >| 7e-01   >| 2e-01      >| 5e-02 <| 1e-02
## <| 3.2   >| 1095        >| 13553   >| 3380       >| 533   <| 18561
## <|       >| 6e-02       >| 7e-01   >| 2e-01      >| 3e-02 <| 4e-02
## <| 3.4   >| 1138        >| 11706   >| 3157       >| 475   <| 16476
## <|       >| 7e-02       >| 7e-01   >| 2e-01      >| 3e-02 <| 4e-02
## <| 3.6   >| 989         >| 10307   >| 2651       >| 427   <| 14374
## <|       >| 7e-02       >| 7e-01   >| 2e-01      >| 3e-02 <| 3e-02
## <| 3.8   >| 962         >| 9801    >| 2433       >| 460   <| 13656
## <|       >| 7e-02       >| 7e-01   >| 2e-01      >| 3e-02 <| 3e-02
## <| 4     >| 823         >| 7914    >| 2060       >| 413   <| 11210
## <|       >| 7e-02       >| 7e-01   >| 2e-01      >| 4e-02 <| 2e-02
## <| 4.2   >| 1306        >| 18628   >| 5601       >| 525   <| 26060
## <|       >| 5e-02       >| 7e-01   >| 2e-01      >| 2e-02 <| 6e-02
```

```
## <| 4.4   >| 1252        >| 18740   >| 5295       >| 547   <| 25834
## <|       >| 5e-02       >| 7e-01   >| 2e-01      >| 2e-02 <| 6e-02
## <| 4.6   >| 1175        >| 18234   >| 4348       >| 564   <| 24321
## <|       >| 5e-02       >| 7e-01   >| 2e-01      >| 2e-02 <| 5e-02
## <| 4.8   >| 1164        >| 17451   >| 4086       >| 570   <| 23271
## <|       >| 5e-02       >| 7e-01   >| 2e-01      >| 2e-02 <| 5e-02
## <| 5     >| 1126        >| 15875   >| 3817       >| 571   <| 21389
## <|       >| 5e-02       >| 7e-01   >| 2e-01      >| 3e-02 <| 5e-02
## <| 5.2   >| 664         >| 16132   >| 5066       >| 256   <| 22118
## <|       >| 3e-02       >| 7e-01   >| 2e-01      >| 1e-02 <| 5e-02
## <| 5.4   >| 863         >| 18390   >| 6099       >| 372   <| 25724
## <|       >| 3e-02       >| 7e-01   >| 2e-01      >| 1e-02 <| 6e-02
## <| 5.6   >| 1297        >| 21335   >| 7584       >| 458   <| 30674
## <|       >| 4e-02       >| 7e-01   >| 2e-01      >| 1e-02 <| 7e-02
## <| 5.8   >| 1313        >| 20871   >| 6810       >| 514   <| 29508
## <|       >| 4e-02       >| 7e-01   >| 2e-01      >| 2e-02 <| 7e-02
## <| 6     >| 1165        >| 17124   >| 5601       >| 456   <| 24346
## <|       >| 5e-02       >| 7e-01   >| 2e-01      >| 2e-02 <| 5e-02
## <| 6.2   >| 116         >| 7367    >| 2839       >| 38    <| 10360
## <|       >| 1e-02       >| 7e-01   >| 3e-01      >| 4e-03 <| 2e-02
## <| 6.4   >| 176         >| 7305    >| 3147       >| 43    <| 10671
## <|       >| 2e-02       >| 7e-01   >| 3e-01      >| 4e-03 <| 2e-02
## <| 6.6   >| 254         >| 8170    >| 3627       >| 60    <| 12111
## <|       >| 2e-02       >| 7e-01   >| 3e-01      >| 5e-03 <| 3e-02
## <| 6.8   >| 467         >| 12345   >| 5433       >| 144   <| 18389
## <|       >| 3e-02       >| 7e-01   >| 3e-01      >| 8e-03 <| 4e-02
## <| 7     >| 549         >| 14864   >| 5362       >| 184   <| 20959
## <|       >| 3e-02       >| 7e-01   >| 3e-01      >| 9e-03 <| 5e-02
## <| Total >| 22671       >| 321185  >| 97316      >| 9789  <| 450961
## |======================================================================
```

```
## Code snippet 1.16
boxplot(pct_carrier_delay ~ airline, data=airline_stats, ylim=c(0,50))
```

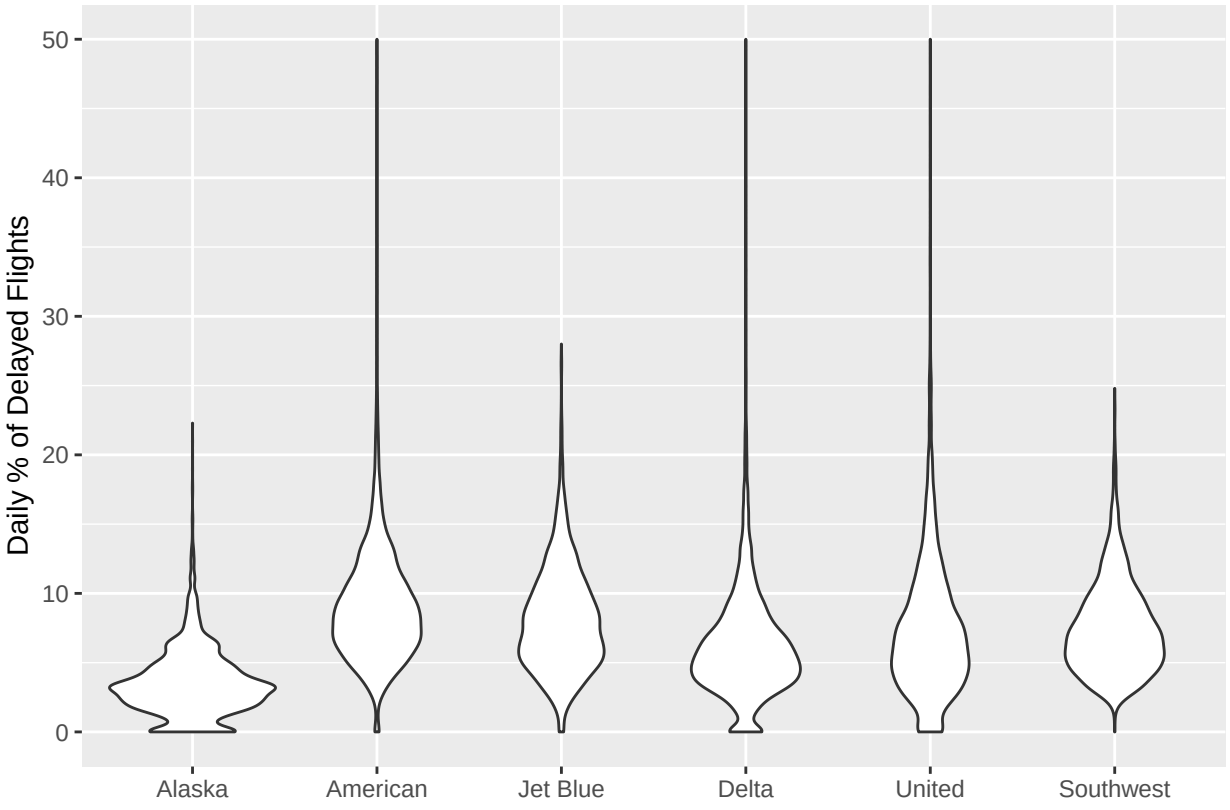



```
## Code for figure 10
png(filename=file.path(PSDS_PATH, "figures", "psds_0110.png"), width = 4, height=4, units='in', res=300)
par(mar=c(4,4,0,0)+.1)
boxplot(pct_carrier_delay ~ airline, data=airline_stats, ylim=c(0,50), cex.axis=.6,
        ylab="Daily % of Delayed Flights")

dev.off()
```

```
## pdf
##   2
```

```
## Code snippet 1.17

ggplot(data=airline_stats, aes(airline, pct_carrier_delay))  +
  ylim(0, 50) +
  geom_violin() +
  labs(x="", y="Daily % of Delayed Flights")
```

```
## Warning: Removed 38 rows containing non-finite values (stat_ydensity).
```




```
## Code for figure 11
png(filename=file.path(PSDS_PATH, "figures", "psds_0111.png"), width = 4, height=4, units='in', res=300)

ggplot(data=airline_stats, aes(airline, pct_carrier_delay)) +
  ylim(0, 50) +
  geom_violin(draw_quantiles = c(.25, .5, .75), linetype=2) +
  geom_violin(fill=NA, size=1.1) +
  theme_bw() +
  labs(x="", y="% of Delayed Flights")
```

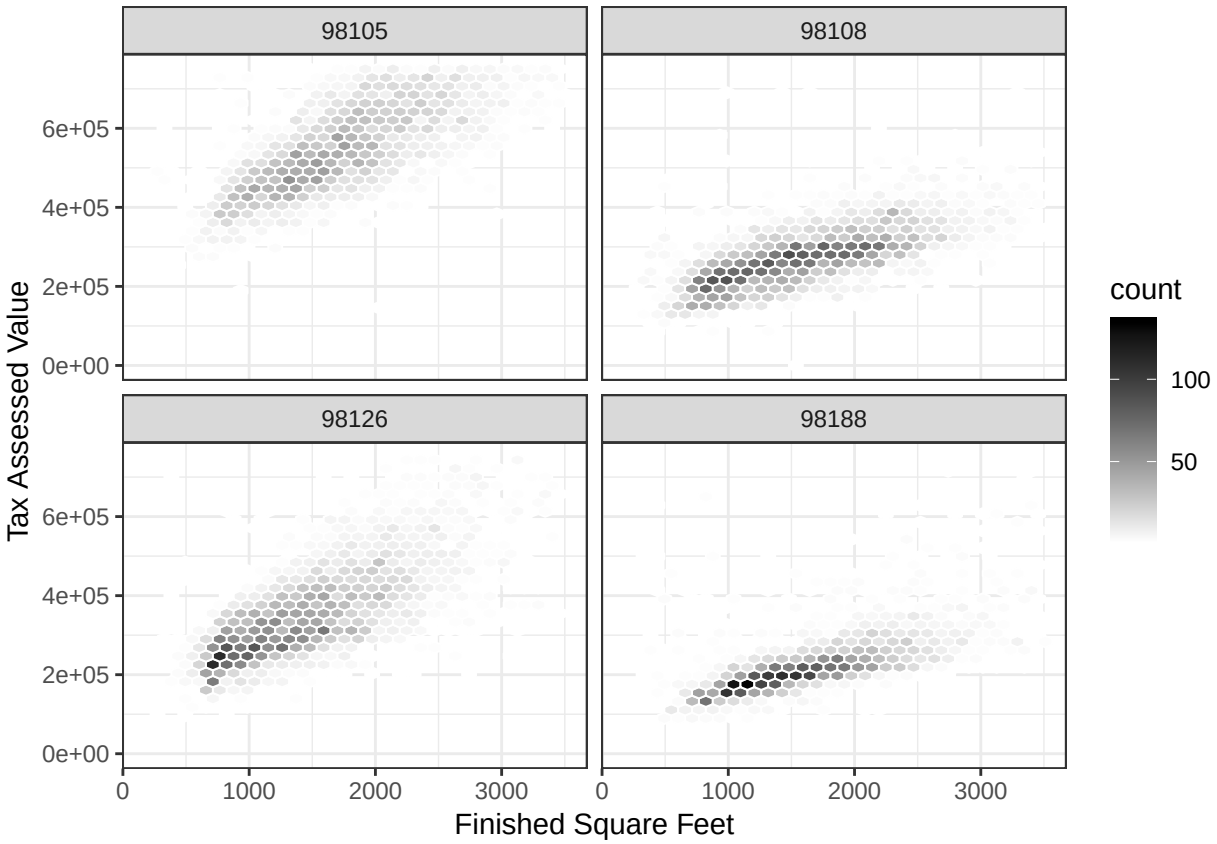
```
## Warning: Removed 38 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 38 rows containing non-finite values (stat_ydensity).
```

```
dev.off()
```

```
## pdf
##   2
```

```
## Code snippet 1.18

ggplot(subset(kc_tax0, ZipCode %in% c(98188, 98105, 98108, 98126)),
         aes(x=SqFtTotLiving, y=TaxAssessedValue)) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient( low="white", high="black") +
  labs(x="Finished Square Feet", y="Tax Assessed Value") +
  facet_wrap("ZipCode")
```




```
## Code for figure 12
png(filename=file.path(PSDS_PATH, "figures", "psds_0112.png"), width = 5, height=4, units='in', res=300)

ggplot(subset(kc_tax0, ZipCode %in% c(98188, 98105, 98108, 98126)),
       aes(x=SqFtTotLiving, y=TaxAssessedValue)) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient( low="gray95", high="blue") +
  labs(x="Finished Square Feet", y="Tax Assessed Value") +
  facet_wrap("ZipCode")
dev.off()
```

```
## pdf
##   2
```